IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Tomohisa SHIGA

International Application No.: PCT/JP03/03716

International Filing Date: March 26, 2003

For: OPERATION-PROCESSING DEVICE, METHOD FOR CONSTRUCTING THE SAME, AND OPERATION-PROCESSING SYSTEM AND METHOD

745 Fifth Avenue
New York, NY 10151

EXPRESS MAIL

Mailing Label Number: EV385414635US

Date of Deposit: September 23, 2004

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Adam Ahmed
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

CLAIM OF PRIORITY UNDER 37 C.F.R. § 1.78(a)(2)

Mail Stop PCT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Pursuant to 35 U.S.C. 119, this application is entitled to a claim of priority to Japan Application No. 2002-088916 and 2002-195123 filed 27 March and 3 July 2002.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP
Attorneys for Applicant

By:
William S. Frommer
Reg. No. 25,506
Tel. (212) 588-0800

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００２年　７月　３日

出願番号
Application Number: 特願２００２－１９５１２３

[ST.10/C]: [ＪＰ２００２－１９５１２３]

出願人
Applicant(s): ソニー株式会社

２００３年　２月１４日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３００７２３４

【書類名】　特許願
【整理番号】　0290181903
【提出日】　平成14年 7月 3日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 15/76
G06F 15/78
H03K 19/173
【発明者】
【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内
【氏名】　志賀　知久
【特許出願人】
【識別番号】　000002185
【氏名又は名称】　ソニー株式会社
【代理人】
【識別番号】　100090376
【弁理士】
【氏名又は名称】　山口　邦夫
【電話番号】　03-3291-6251
【選任した代理人】
【識別番号】　100095496
【弁理士】
【氏名又は名称】　佐々木　榮二
【電話番号】　03-3291-6251
【手数料の表示】
【予納台帳番号】　007548
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1

出証特２００３－３００７２３４

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9709004

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　データ処理システム、データ処理装置及びデータ処理方法

【特許請求の範囲】

【請求項１】　一方で、所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理するシステムであって、

前記レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成するプログラム作成装置と、

前記プログラム作成装置で作成された圧縮プログラムを取得してレジスタ種類を解読し、前記レジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するデータ処理装置とを備えることを特徴とするデータ処理システム。

【請求項２】　前記データ処理装置は、

複数のレジスタと、

前記レジスタを指定するための圧縮プログラムを記憶する記憶手段と、

前記記憶手段から圧縮プログラムを読み出してレジスタ種類を解読し、当該レジスタ種類に基づいて前記レジスタを指定するための命令ビット数を復元する命令解読復元手段と、

前記命令解読復元手段によって復元された所定の長さの命令に基づいて前記レジスタを指定して任意の演算を実行する命令実行演算手段とを有することを特徴とする請求項１に記載のデータ処理システム。

【請求項３】　前記レジスタ種類は、

Ｎ個のレジスタを使用する場合であって、

前記Ｎ個のレジスタに第１番から第Ｎ番のシリアル番号を付与したとき、

前記第１番から第Ｋ番のグループのレジスタを使用頻度が高い部類に、

前記第Ｋ＋１番から第Ｎ番のグループのレジスタを使用頻度が低い部類に分けることを特徴とする請求項１に記載のデータ処理システム。

【請求項４】　レジスタを使用する頻度に基づいて当該レジスタを指定する

命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムに基づいてデータを処理する装置であって、

複数のレジスタと、

前記レジスタを指定するための圧縮プログラムを記憶する記憶手段と、

前記記憶手段から圧縮プログラムを読み出してレジスタ種類を解読し、当該レジスタ種類に基づいて前記レジスタを指定するための命令ビット数を復元する命令解読復元手段と、

前記命令解読復元手段によって復元された所定長さの命令に基づいて前記レジスタを指定して任意の演算を実行する命令実行演算手段とを備えることを特徴とするデータ処理装置。

【請求項５】　前記レジスタ種類は、

Ｎ個のレジスタを使用する場合であって、

前記Ｎ個のレジスタに第１番から第Ｎ番のシリアル番号を付与したとき、

前記第１番から第Ｋ番のグループのレジスタを使用頻度が高い部類に、前記第Ｋ＋１番から第Ｎ番のグループのレジスタを使用頻度が低い部類に分けられることを特徴とする請求項４に記載のデータ処理装置。

【請求項６】　プログラム作成系で所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用してデータを処理する方法であって、

前記プログラム作成系では、

前記レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成し、

前記プログラム実行系では、

前記プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、

解読された前記レジスタ種類に基づいて当該レジスタを指定する命令ビット数

を復元し、

復元された所定長さの前記命令に基づいて複数のレジスタを指定して任意の演算を実行することを特徴とするデータ処理方法。

【請求項７】　前記レジスタ種類は、

Ｎ個のレジスタを使用する場合であって、

前記Ｎ個のレジスタ第１番から第Ｎ番のシリアル番号を付与したとき、

前記第１番から第Ｋ番のグループのレジスタを使用頻度が高い部類に、

前記第Ｋ＋１番から第Ｎ番のグループのレジスタを使用頻度が低い部類に分けることを特徴とする請求項６に記載のデータ処理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、システムプログラムに基づいて各種データ処理をする中央演算装置（ＣＰＵ）やマイクロプロセッサユニット（ＭＰＵ）等、また、プログラム可能な論理演算素子（ＰＬＤ）や、これらの組み込み電子機器等に適用して好適なデータ処理システム、データ処理装置及びデータ処理方法に関する。

【０００２】

詳しくは、複数のレジスタを指定して任意の演算を実行するデータ処理装置を備え、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定の命令長の命令構造を有するプログラムを復元するようにして、プログラムデータを格納するＲＯＭ等のメモリ容量を低減できるようにすると共に、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減できるようにしたものである。

【０００３】

【従来の技術】

近年、携帯端末装置や、電子カード、情報処理装置等の各種電子機器にＣＰＵ（中央演算処理装置）を含むマイクロプロセッサが使用される場合が多くなって

きた。この種のプロセッサには命令実行演算部の他に命令実行プログラムを格納するための読み出し専用メモリ（以下、ＲＯＭという）や、命令実行演算処理に使用される多くのレジスタ等が実装されている。

【０００４】

従来方式のマイクロプロセッサによれば、任意の電子機器に当該プロセッサを組み込んで用いる場合、ある動作を行う命令はその動作と、その動作を行うための命令とが一対一に対応していた。つまり、使用頻度が高いレジスタも、使用頻度が低いレジスタも、当該レジスタを指定するための命令ビット数を同等にして一律の長さの命令により作成されたプログラムが使用される場合が多い。このことで、固定長の命令がＲＯＭに格納されて使用される。

【０００５】

他方で、半導体集積回路技術の発展により、多大な数のレジスタをプロセッサ（以下データ処理装置ともいう）内に実装することが可能になってきた。この場合、レジスタを特定するための命令ビット数も益々多く必要になる。例えば、１０２４個のレジスタが実装される場合、１０２４個のレジスタ中でその１つを特定するためには、命令ビットとして１０ビットが必要となる。しかし、実際のプログラムでは全てのレジスタへのアクセス頻度は一様ではなく、アクセス頻度に斑がある。頻繁にアクセスされるレジスタの番号は一般的にコンパイラによって決定される。

【０００６】

【発明が解決しようとする課題】

ところで、従来方式のデータ処理装置によれば、命令実行プログラムを格納するＲＯＭを実装するに当たり以下のような問題がある。

【０００７】

①　命令実行プログラムのコード全体を眺めてみると、レジスタを指定するための例えば、１０ビットの命令ビット内、この１０ビット全体が一様に使用されることが少ない。従って、命令実行プログラムを格納するメモリ中（例えばＲＯＭやフラッシュメモリ）に無駄なビットが多く存在してしまう。これにより、全てのレジスタを単一の命令ビット数で表現する方法は効率良くＲＯＭを使用する

ことに関して妨げとなる。

【０００８】

② また、メモリセルや論理演算素子から成るプログラマブル・ロジック・デバイス（Programmable Logic Device；ＰＬＤ）によりマイクロプロセッサ等を構築しようとした場合に、同一半導体チップ上に命令実行演算部を配置し、その周辺部にレジスタアレイや、ＲＯＭ等を配置する方法が考えられる。この場合に、プロセッサの多機能化の要求から命令実行プログラムが多くなると、このプログラムを格納するＲＯＭのメモリ容量が多く必要になる。従って、メモリセルがＲＯＭ構築に占有されてしまい、多くのメモリセルをレジスタに割り当てることが困難になる。

【０００９】

そこで、この発明はこのような従来の課題を解決したものであって、レジスタの使用頻度に応じて命令の長さを可変できるようにすると共に、頻繁にアクセスするレジスタには短い長さの命令をセットできるようにし、プログラムデータを格納するＲＯＭ等のメモリ容量を低減できるようにしたデータ処理システム、データ処理装置及びデータ処理方法を提供することを目的とする。

【００１０】

【課題を解決するための手段】

上述した課題は、一方で、所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理するシステムであって、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成するプログラム作成装置と、このプログラム作成装置で作成された圧縮プログラムを取得してレジスタ種類を解読し、レジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するデータ処理装置とを備えることを特徴とするデータ処理システムによって解決される。

【００１１】

本発明に係るデータ処理システムによれば、一方で、所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理する場合に、プログラム作成装置ではレジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムが作成される。

【００１２】

データ処理装置では、プログラム作成装置で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算が実行される。

【００１３】

従って、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。これにより、プログラム実行系ではＲＯＭ等に圧縮した命令をセットすることができ、プログラムデータを格納するＲＯＭ等のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

【００１４】

本発明に係るデータ処理装置は、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムに基づいてデータを処理する装置であって、複数のレジスタと、レジスタを指定するための圧縮プログラムを記憶する記憶手段と、この記憶手段から圧縮プログラムを読み出してレジスタ種類を解読し、当該レジスタ種類に基づいてレジスタを指定するための命令ビット数を復元する命令解読復元手段と、この命令解読復元手段によって復元された所定長さの命令に基づいてレジスタを指定して任意の演算を実行する命令実行演算手段とを備えることを特徴とするものである。

【００１５】

本発明に係るデータ処理装置によれば、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムに基づいてデータを処理する場合に、記憶手段には複数のレジスタ中から当該レジスタを指定するための圧縮プログラムが記憶される。命令解読復元手段では、この記憶手段から圧縮プログラムを読み出してレジスタ種類が解読され、このレジスタ種類に基づいて当該レジスタを指定するための命令ビット数が復元される。これを前提にして、命令実行演算手段では命令解読復元手段によって復元された所定の命令長のプログラムに基づいてレジスタを指定して任意の演算を実行するようになされる。

【００１６】

従って、レジスタの使用頻度に応じて可変された命令の長さの圧縮プログラムであって、頻繁にアクセスするレジスタには短い長さの命令がセットされた、圧縮プログラムデータを記憶手段に格納することができるので、そのメモリ容量を低減することができる。これにより、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

【００１７】

本発明に係るデータ処理方法はプログラム作成系で所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用してデータを処理する方法であって、プログラム作成系では、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成し、プログラム実行系では、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、ここで解読されたレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、ここで復元された所定長さの命令に基づいて複

数のレジスタを指定して任意の演算を実行することを特徴とするものである。

【００１８】

本発明に係るデータ処理方法によれば、プログラム作成系で所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用してデータを処理する場合に、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。

【００１９】

従って、プログラム実行系ではＲＯＭ等に圧縮した命令をセットすることができ、プログラムデータを格納するＲＯＭ等のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

【００２０】

【発明の実施の形態】

続いて、この発明に係るデータ処理システム、データ処理装置及びデータ処理方法の一実施の形態について、図面を参照しながら説明をする。

【００２１】

（１）実施形態

図１は本発明に係る実施形態としてのデータ処理システム１００の構成例を示すブロック図である。

この実施形態では複数のレジスタを指定して任意の演算を実行するデータ処理装置を備え、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定の命令長の命令構造を有するプログラムを復元するようにして、プログラムデータを格納するＲＯＭ等のメモリ容量を低減できるようにすると共に、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減できるようにしたもの

である。

【００２２】

また、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるようにすると共に、頻繁にアクセスするレジスタには短い長さの命令をセットできるようにする。

【００２３】

図１に示すデータ処理システム１０は、一方で、所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理するシステムである。データ処理システム１０ではプログラム作成系Ｉを成すプログラム作成装置２００が準備される。新規に設計製造されるデータ処理装置１００を動作させるためのプログラムを作成するためである。データ処理装置１００はプログラム実行系IIを構成し、当該装置１００内には命令実行演算手段や、記憶手段、複数のレジスタ等が実装される。

【００２４】

プログラム作成装置２００では、データ処理装置１００でレジスタを使用する頻度に基づいて当該レジスタを指定するための命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムＡＰを作成するようになされる。データ処理装置１００に実装されるプログラム格納用の記憶手段のメモリ容量を削減するためである。

【００２５】

プログラム作成装置２００は例えば、データベース２１、キーボード２２、マウス２３、表示装置２４及び制御装置２５を有している。データベース２１にはデータ処理装置１００のプログラム作成に必要なデータが格納される。例えば、Ｃ言語によるプログラムの記述に必要な「Ｇｌｏｂａｌ変数宣言」、「関数宣言」、「Ｌｏｃａｌ変数宣言」、「代入」、「加算」、「比較」及び「分岐」が格納される。データベース２１には制御装置２５が接続されており、この制御装置２５にはキーボード２２、マウス２３及び表示装置２４が接続されている。

【００２６】

プログラム作成装置２００では表示装置２４にＣ言語によるプログラム記述画面を表示して、キーボード２２及びマウス２３を使用してプログラムが作成される。例えば、新規な設計製造に係るデータ処理装置１００がＮ個のレジスタを使用する場合であって、Ｎ個のレジスタに第１番から第Ｎ番のシリアル番号を付与したとき、第１番から第Ｋ番のグループのレジスタを使用頻度が高い部類として「Ｌｏｃａｌ変数宣言」がなされ、第Ｋ＋１番から第Ｎ番のグループのレジスタを使用頻度が低い部類として「Ｇｌｏｂａｌ変数宣言」がなされる。

【００２７】

これらの宣言はキーボード２２や、マウス２３を使用して指定され、これはレジスタ種類を２つに分類して使用頻度が高いレジスタは短い命令ビット数で命令セットし、使用頻度が低いレジスタは長い命令ビット数で命令セットするためである。データ処理装置１００においてレジスタの数が例えば、４千～８千個程度になると、これを指定する命令ビット数が１２乃至１３ビット必要になる。

【００２８】

制御装置２５ではレジスタの使用頻度に応じて命令の長さを可変するようになされる。頻繁にアクセスするレジスタは短い長さの命令をセットするためである。使用頻度が高いレジスタは短い命令ビット数が割り当てられ、使用頻度が低いレジスタは長い命令ビット数が割り当てられる。

【００２９】

データ処理装置１００ではこのプログラム作成装置２００で作成された圧縮プログラムＡＰを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するようになされる。

【００３０】

データ処理装置１００は例えば、命令解読復元手段３、記憶手段４、レジスタアレイ１１及び命令実行演算手段５０を有している。レジスタアレイ１１は複数のレジスタを集合したものである。

【００３１】

記憶手段４にはレジスタアレイ１１の中から該当レジスタを指定するための圧

縮プログラムＡＰが格納される。圧縮プログラムＡＰはプログラム作成装置２００で作成されたものが使用される。例えば、圧縮プログラムＡＰはデータ処理装置１００で構築されたプログラム格納用の記憶手段４に、ＲＯＭライタ等を使用して書き込まれる。

【００３２】

これはデータ処理装置１００に関して、複数のメモリセルや算術論理素子により構成されるプログラマブル・ロジック・デバイス（Programmable Logic Device；ＰＬＤ）からプロセッサを構築する場合があるからである。ＲＯＭとして機能させるメモリセルの占有率を低減することができる。もちろん、プログラム格納用の記憶手段４を演算処理装置とは別個に製造し、個々の記憶手段４に圧縮プログラムＡＰを格納してから同一基板上に実装する方法であってもよい。記憶手段４として読み出し専用のメモリ（ＲＯＭ）や、ＥＥＰＲＯＭ（フラッシュメモリ）が使用されるからである。

【００３３】

記憶手段４には命令解読復元手段３が接続されており、この記憶手段４から圧縮プログラムＡＰを読み出してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定するための命令ビット数を復元するようになされる。命令長を揃え、この命令に基づいて複数のレジスタを指定するためである。

【００３４】

命令解読復元手段３には命令実行演算手段５０が接続されており、この命令実行演算手段５０にはレジスタアレイ１１が接続されている。命令実行演算手段５０では命令解読復元手段３によって復元された所定の命令長のプログラムに基づいてレジスタアレイ１１内で該当レジスタを指定して任意の演算を実行するようになされる。

【００３５】

続いて、本発明に係るデータ処理方法について、当該データ処理システム１０における処理例について説明をする。図２はデータ処理システム１０における処理例を示すフローチャートである。

【００３６】

このシステム１０ではプログラム作成系Ｉで所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、プログラム実行系IIで当該プログラムと複数のレジスタとを使用してデータを処理する場合を前提とする。このデータ処理装置１００がＮ個のレジスタを使用する場合であって、Ｎ個のレジスタに第１番から第Ｎ番のシリアル番号が付与される場合を例にとる。

【００３７】

これを処理条件にして、プログラム作成系では図２ＡにフローチャートのステップＡ１で所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集する。そして、ステップＡ２でレジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮し、命令長を短くする。例えば、第Ｋ＋１番から第Ｎ番のグループのレジスタを指定する命令ビット数をｎビットとし、第１番から第Ｋ番のグループのレジスタを指定する命令ビット数をｍビットとしたとき、例えば、ｎ－ｍ＝８ビットとなるように、第１番から第Ｋ番のグループのレジスタを指定する命令ビット数が圧縮される。

【００３８】

その後、ステップＡ３で当該プログラムの命令構造の中にレジスタ種類を記述する。例えば、被数及び加数を保持する１組のレジスタ、これを「レジスタ番号１」のレジスタの種類を「レジスタ種類１」、及び「レジスタ番号２」のレジスタの種類を「レジスタ種類２」としたとき、第１番から第Ｋ番のグループのレジスタに関して使用頻度が高い場合は「レジスタ種類１」及び「レジスタ種類２」にコード「０」が記述される。また、第Ｋ＋１番から第Ｎ番のグループのレジスタに関して使用頻度が低い場合は「レジスタ種類１」及び「レジスタ種類２」にコード「１」が記述される。

【００３９】

そして、ステップＡ４で命令長の異なる圧縮プログラムＡＰを作成する。この圧縮プログラムＡＰにおいて、第１番から第Ｋ番のグループのレジスタを指定する命令ビット数に関してはｍビットであり、第Ｋ＋１番から第Ｎ番のグループのレジスタを指定する命令ビット数に関してはｎビットである。上述の例で第１番

から第Ｋ番のグループのレジスタ指定を含む命令形態では、第Ｋ＋１番から第Ｎ番のグループのレジスタ指定を含む命令形態に比べて命令長が１６ビット短くなる。

【００４０】

一方、プログラム実行系ではプログラム作成系で作成された圧縮プログラムＡＰを図２Ｂに示すフローチャートのステップＢ１で取得する。例えば、圧縮プログラムＡＰはデータ処理装置１００で構築されたプログラム格納用の記憶手段４に、ＲＯＭライタ等を使用して書き込まれる。この圧縮プログラムＡＰでは使用頻度が高いレジスタは短い命令ビット数＝ｍビットで命令セットされ、使用頻度が低いレジスタは長い命令ビット数＝ｎビットで命令セットされている。

【００４１】

そして、ステップＢ２で命令を実行するかを判断する。この際の判断は周知技術によりなされる。命令を実行する場合はステップＢ３でレジスタ種類を解読する。例えば、「レジスタ種類１」及び「レジスタ種類２」に関してコード「０」から使用頻度が高いレジスタとして第１番から第Ｋ番のグループのレジスタ番号が解読され、「レジスタ種類１」及び「レジスタ種類２」に関してコード「１」から使用頻度が低いレジスタとして第Ｋ＋１番から第Ｎ番のグループのレジスタ番号が解読される。

【００４２】

そして、解読されたレジスタ種類に基づいてステップＢ４で当該レジスタを指定する命令ビット数を復元する。例えば、第１番から第Ｋ番のグループのレジスタの命令ビット数＝ｍビットの上位、この例で上位８ビットに「０」が付加される。第１番から第Ｋ番のグループのレジスタの命令ビット数が、第Ｋ＋１番から第Ｎ番のグループのレジスタの命令ビット数と同様にしてｎビットに揃えられる。

【００４３】

ここで復元された所定長さの命令に基づいてステップＢ５で複数のレジスタを指定して任意の演算を実行する。その後、ステップＢ６で演算処理を終了するかを判断する。演算処理を終了しない場合はステップＢ２に戻って命令を実行する

かを判断して演算処理を継続する。演算処理を終了する場合は電源オフ情報等を検出して当該演算処理を終了する。

【００４４】

このように、本発明に係る実施形態としてのデータ処理システム１０によれば、一方で、所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理する場合に、プログラム作成装置２００ではレジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムＡＰが作成される。

【００４５】

データ処理装置１００では、プログラム作成装置２００で作成された圧縮プログラムＡＰを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算が実行される。

【００４６】

従って、プログラム作成系Ｉではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。これにより、プログラム実行系IIではＲＯＭ等の記憶手段４に圧縮した命令をセットすることができ、プログラムデータを格納する記憶手段４のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができるようになる。

【００４７】

（２）実施例

図３は本発明に係る実施例としてのマイクロプロセッサ１０１の構成例を示すブロック図である。

この実施例ではデータ処理装置１００に外部メモリ２を接続してマイクロプロ

セッサ１０１を構成し、複数のレジスタを指定して任意の演算を実行する。そのために、プログラム作成系Ｉで作成された機械語の命令の圧縮プログラムＡＰを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定の命令長の命令構造を有するプログラムを復元するようにした。そうすることでプログラムデータを格納するＲＯＭ等のメモリ容量を低減できるようにしたものである。

【００４８】

図３に示すマイクロプロセッサ１０１は、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムＡＰに基づいてデータを処理する装置である。このプロセッサ１０１ではプログラム作成装置２００で作成された圧縮プログラムＡＰを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するようになされる。

【００４９】

マイクロプロセッサ１０１は例えば、レジスタアレイ１１、命令ビット復元デコーダ１３、ＲＯＭ１４、及び命令実行演算手段５０を有している。レジスタアレイ１１は複数のレジスタを集合したものである。レジスタアレイ１１には例えば、８１９２個×３２ｂｉｔのレジスタｒｉ（ｉ＝０～８１９１）が設けられる。各々のレジスタｒｉは書込みアドレスＡｗ及び書込み制御信号Ｓｗに基づいて任意の値を保持し、及び、読出しアドレスＡｒに基づいて被数Ｘや加数Ｙ等の値を出力するようになされる。

【００５０】

このマイクロプロセッサ１０１は記憶手段の一例となるプログラム格納用のＲＯＭ１４が実装されており、レジスタアレイ１１の中から該当レジスタｒｉを指定するための圧縮プログラムＡＰが格納される。圧縮プログラムＡＰは機械語の命令（Ｉnstruction）構造を有しており、プログラム作成装置２００で作成されたものが使用される。例えば、圧縮プログラムＡＰはＲＯＭライタ等を使用して

ＲＯＭ１４に書き込まれる。命令実行時、ＲＯＭ１４は例えば、プログラムカウンタ５４からのカウント出力信号Ｓ５に基づいて圧縮プログラムＡＰを出力するようになされる。

【００５１】

ＲＯＭ１４には命令ビット復元デコーダ１３が接続されており、このＲＯＭ１４から機械語の命令の圧縮プログラムＡＰを読み出して命令制御信号Ｓ４、命令信号Ｓ９及び各引数信号Ｓ１０を発生するようになされる。命令信号Ｓ９には１ｏａｄ命令、ａｄｄ命令、ｃｍｐ命令、ｊｕｍｐ命令が含まれる。各引数信号Ｓ１０にはアクセス方法＃１，アクセス方法＃２，「レジスタ種類１」，「レジスタ種類２」，レジスタ番号ｒ０，ｒ１・・・等、フラグ状態（ｆｌａｇ　ｃｏｎｄｉｔｉｏｎ）及びジャンプアドレス等が含まれる。

【００５２】

この圧縮プログラムＡＰには、レジスタ相対メモリアドレッシング処理を実行するための演算命令を含んでいる。この処理では演算命令に基づいて一のレジスタを選択し、ここで選択されたレジスタが保持する値によって外部メモリ２を選択するようになされる。この処理は例えば、アクセス方法＃１によって実行される。

【００５３】

この例で命令ビット復元デコーダ１３は、「レジスタ種類１」及び「レジスタ種類２」を解読し、この「レジスタ種類１」及び「レジスタ種類２」に基づいて当該レジスタｒｉを指定するための命令ビット数＝ｎビットを復元するようになされる。命令長を圧縮前のビット数に揃え、この命令に基づいて複数のレジスタｒｉ等を指定するためである。上述の命令制御信号Ｓ４は命令読出しステートマシーン５２に出力される。

【００５４】

上述のレジスタアレイ１１及び命令ビット復元デコーダ１３には命令実行演算手段５０が接続されている。命令実行演算手段５０では命令ビット復元デコーダ１３によって復元された所定の命令長のプログラムに基づいてレジスタアレイ１１内で該当レジスタｒｉを指定して任意の演算を実行するようになされる。

【００５５】

命令実行演算手段５０は、算術論理演算ユニット（Arithmetic and Logic Unit：以下でＡＬＵという）１２、実行ステートマシーン５１、命令読出しステートマシーン５２、セレクタ５３、プログラムカウンタ（ＰＣ）５４、＋１インクリメンタ５５、入力用のセレクタ５９、ラッチ回路５８、５１０、５１１を有しており、レジスタ相対メモリアドレッシング処理を実行するようになされる。

【００５６】

レジスタアレイ１１にはデータ信号線Ｌ２０が接続されており、この信号線Ｌ２０を通じてＡＬＵ１２が接続されている。ＡＬＵ１２ではレジスタアレイ１１の中で指定されたレジスタから読み出されたＸやＹ等の値を演算するようになされる。演算結果の値はＺである。演算種目は足し算、かけ算、引き算、わり算等である。演算種目は実行ステートマシーン５１から出力されるＡＬＵ制御信号Ｓ３５に基づいて設定される。データ信号線Ｌ２０には、ＡＬＵ１２の他にラッチ回路５８、５１１、５９等が接続されている。データ信号線Ｌ２０にはＤＡＴＡ、被数Ｘ値、加数Ｙ値等が伝送される。

【００５７】

命令ビット復元デコーダ１３には実行ステートマシーン５１及び命令読出しステートマシーン５２が接続されており、命令ビット復元デコーダ１３によって解読された演算命令を実行するためにレジスタアレイ１１及びＡＬＵ１２を制御するようになされる。

【００５８】

命令読出しステートマシーン５２では命令ビット復元デコーダ１３から出力される命令制御信号Ｓ４に基づいてプログラムカウンタ５４及び実行ステートマシーン５１を制御する。例えば、当該マシーン５２は命令ビット復元デコーダ１３から命令信号Ｓ９及び各引数信号Ｓ１０が実行ステートマシーン５１へ出力されると共に命令実行開始信号Ｓ２９を出力する。

【００５９】

実行ステートマシーン５１にはレジスタアレイ１１、ＡＬＵ１２、ラッチ回路

５８，５１０、５１１及びセレクタ５９が接続されている。当該マシーン５１では命令実行開始信号Ｓ２９に基づいて命令の実行を開始する。例えば、データの書込み時には、書込み制御信号Ｓｗがレジスタアレイ１１に出力され、セレクタ５９には選択制御信号Ｓ２４が出力される。データの読出し時には、レジスタアレイ１１に読出しアドレスＡｒが出力される。

【００６０】

演算時には、ラッチ制御信号Ｓ３４がラッチ回路５８に出力され、ラッチ回路５１０にはラッチ制御信号Ｓ３８が出力される。当該プロセッサ外部には外部制御信号Ｓ１６が出力される。命令の実行が終了すると、実行ステートマシーン５１は命令読み出しステートマシーン５２へ実行終了信号Ｓ２６を出力し、プログラムカウンタ５４の値を進めるようになされる。

【００６１】

この実行ステートマシーン５１及び命令読み出しステートマシーン５２にはセレクタ５３が接続されており、選択制御信号Ｓ２８に基づいてインクリメント出力信号Ｓ７又は分岐制御信号Ｓ２７のいずれか一方を選択し、これをセレクタ出力としてプログラムカウンタ５４に出力するようになされる。選択制御信号Ｓ２８は実行ステートマシーン５１から供給される。インクリメント出力信号Ｓ７はインクリメンタ５５からセレクタ５３へ出力される。

【００６２】

プログラムカウンタ５４ではカウント制御信号Ｓ３０に基づいてＲＯＭ１４から圧縮プログラムＡＰを読み出す場所が指定される。＋１インクリメンタ５５はプログラムカウンタ５４のカウント出力信号Ｓ５を「＋１」してインクリメンとするようになされる。カウント制御信号Ｓ３０は命令解読ステートマシーン５２から供給される。このカウント出力信号Ｓ５は＋１インクリメンタ５５の他にＲＯＭ１４に出力される。

【００６３】

セレクタ５９はデータバス１９Ａ、レジスタアレイ１１及びＡＬＵ１２に接続されており、データバス１９Ａから取り込んだデータ（ＤＡＴＡ）、レジスタアレイ１１から出力される被数Ｘ値（加数Ｙ値）又はＡＬＵ１２から出力される演

算結果値Ｚのいずれかを選択制御信号Ｓ２４に基づいて入力制御するようになされる。

【００６４】

ラッチ回路５８はレジスタアレイ１１の読出しポートとＡＬＵ１２の間に接続されており、ラッチ制御信号Ｓ３４に基づいてレジスタｒｉの出力値Ｘをラッチするようになされる。ラッチ回路５１０はＡＬＵ１２の比較出力部等に接続されており、ラッチ制御信号Ｓ３８に基づいて一致検出信号Ｓ２２をラッチして、フラグ状態（ｆｌａｇ　ｃｏｎｄｉｔｉｏｎ）信号Ｓ２３を出力するようになされる。ラッチ回路５１１はレジスタアレイ１１の読出しポートとアドレスバス１９Ｂとの間に接続されており、ラッチ制御信号Ｓ１７に基づいて外部アドレス（ａｄｄｒｅｓｓ）をラッチするようになされる。

【００６５】

なお、実行された命令によってジャンプ（命令分岐）が発生した場合は、ジャンプ先のアドレスを示す分岐制御信号Ｓ２７を実行ステートマシン５１からセレクタ５３へ出力される。セレクタ５３では選択制御信号Ｓ２８に基づいてその分岐制御信号Ｓ２７を選択し、この分岐制御信号Ｓ２７をプログラムカウンタ５４へ書き込むようになされる。

【００６６】

また、実行ステートマシーン５１、セレクタ５９、データ信号線Ｌ２０及びラッチ回路５１１にはＩ／Ｏインタフェース６０を通じて外部メモリ２が接続されている。レジスタ相対メモリアドレッシング処理に基づいてＡＬＵ１２を動作させるためである。Ｉ／Ｏインタフェース６０と外部メモリ２との間はデータバス１９Ａ、アドレスバス１９Ｂ及びコントロールバス１９Ｃによって接続され、データバス１９Ａによってデータが転送され、アドレスバス１９Ｂによってアドレスが転送され、コントロールバス１９Ｃによって外部制御信号Ｓ１６が外部メモリ２へ転送される。外部メモリ２を制御するためである。外部メモリ２には例えば、５１２Ｍバイト×３２ビットのＲＡＭ（随時書き込み読み出し可能なメモリ）が使用される。

【００６７】

図４はレジスタアレイ１１の内部構成例を示すブロック図である。図４に示すレジスタアレイ１１によれば、例えば、８１９２個の３２ｂｉｔのレジスタｒｉ（ｉ＝０～８１９１）が備えられ、各々のレジスタｒｉの入力には書込みポート１５が接続されている。１ｂｉｔのレジスタはＤ型のフリップ・フロップ回路等から構成される。

【００６８】

書込みポート１５は図３に示したセレクタ５９に接続されており、書込み制御信号Ｓｗ及び書込みアドレスＡｗに基づいて、データバス１９Ａから取り込んだデータ（ＤＡＴＡ）、レジスタアレイ１１から出力される被数Ｘ値（加数Ｙ値）又はＡＬＵ１２から出力される演算結果値Ｚのいずれかをレジスタｒ０～ｒｉに書き込むようになされる。書込みポート１５は実行ステートマシーン５１に接続され、書込みアドレスＡｗを供給するようになされる。

【００６９】

各々のレジスタｒｉの出力には読出しポート１６が接続されている。読出しポート１６はデータ信号線Ｌ２０を通じて図３に示したＡＬＵ１２、ラッチ回路５８、５１１、セレクタ５９等に接続されており、読出しアドレスＡｒに基づいて指定されたレジスタｒｉからデータ（ＤＡＴＡ）を読み出すようになされる。読出しポート１６は実行ステートマシーン５１に接続され、読出しアドレスＡｒを供給するようになされる。

【００７０】

次に、ＲＯＭ１４にセットされる命令の構造例について説明する。図５Ａ～Ｅはマイクロプロセッサ１０１で取り扱う命令の構造例を示すフォーマットである。図６Ａ～Ｄは命令構造における記述内容例を示す表図である。

【００７１】

このＲＯＭ１４にセットされる命令形態は、図５Ａ～Ｅに示すように、＃Ｆ１～＃Ｆ５の５種類である。マイクロプロセッサ１０１では命令形態＃Ｆ２～＃Ｆ４の命令を命令形態＃Ｆ１の命令の形態に復元して取り扱われる。これらの命令形態＃Ｆ１～＃Ｆ５で各々の命令はｌｏａｄ、ａｄｄ及びｃｍｐ命令と、ｊｕｍｐ命令とに大きく二つに分かれる。命令形態＃Ｆ１～＃Ｆ４において、ｌｏａｄ

命令の場合は図６Ａに示すように命令にコード「０」が記述され、ａｄｄ命令の場合は命令にコード「１」が記述され、ｃｍｐ命令の場合は命令にコード「２」が各々記述される。

【００７２】

命令形態＃Ｆ５はｊｕｍｐ命令であって、この場合は命令にコード「３」が記述される。ｃｍｐ命令に関しては比較結果が同じであった場合は、図３に示したラッチ回路５１０のフラグ状態信号Ｓ２３に基づいてｚｅｒｏ　ｆｌａｇが「１」にセットされ、同じでなかった場合は「０」がセットされる。

【００７３】

この例では、ｌｏａｄ、ａｄｄ及びｃｍｐ命令はアクセスするレジスタｒｉの番号が例えば、５ビットで表現できるときは５ビット、それ以外は１３ビットで表現される。つまり、使用頻度が高い第０番のレジスタｒ０から第３１番のレジスタｒ３１をｍ＝５ビットで表現する。使用頻度が低い第３２番のレジスタｒ３２から第８１９１番のレジスタｒ８１９１をｍ＝１３ビットで表現するようになされる。

【００７４】

命令形態＃Ｆ１では命令長が３２ビットであり、「レジスタ番号１」で示されるレジスタｒｉの命令ビット数はｎ＝１３ビットであり、「レジスタ番号２」で示されるレジスタｒｉの命令ビット数もｎ＝１３ビットである。命令形態＃Ｆ２では命令長が２４ビットであり、「レジスタ番号１」で示されるレジスタｒｉの命令ビット数はｍ＝５ビットであり、「レジスタ番号２」で示されるレジスタｒｉの命令ビット数はｎ＝１３ビットである。

【００７５】

また、命令形態＃Ｆ３でも命令長が２４ビットであり、「レジスタ番号１」で示されるレジスタｒｉの命令ビット数はｎ＝１３ビットであり、「レジスタ番号２」で示されるレジスタｒｉの命令ビット数はｍ＝５ビットである。命令形態＃Ｆ４は命令長が１６ビットであり、「レジスタ番号１」で示されるレジスタｒｉの命令ビット数はｍ＝５ビットであり、「レジスタ番号２」で示されるレジスタｒｉの命令ビット数もｍ＝５ビットである。

【００７６】

いずれの命令形態＃Ｆ１～＃Ｆ４においても、最初の２ビットは命令の種類を示している。命令種類に関して、ｌｏａｄは転送、ａｄｄは加算、ｃｍｐは比較、ｊｕｍｐは制御移行（分岐）を各々示している。ｌｏａｄ、ａｄｄ、ｃｍｐ命令の場合は、命令の後に２ビットのアクセス方法＃１、＃２が続く。Ｏｐｅｒａｎｄは左がアクセス方法＃１、「レジスタ番号１」で表され、右がアクセス方法＃２、「レジスタ番号２」で表される。

【００７７】

つまり、アクセス方法＃１は「レジスタ番号１」で示されるレジスタｒｉのアクセス方法を示しており、アクセス方法＃２は「レジスタ番号２」で示されるレジスタｒｉのアクセス方法を示している。アクセス方法＃１とアクセス方法＃２はそれぞれレジスタ番号Ｎｏ．１、レジスタ番号Ｎｏ．２に対応し、これらの間で処理が行われる。いずれも、図６Ｂに示すようにアクセス方法＃１、＃２には２種類が準備されている。

【００７８】

アクセス方法＃１、＃２に関してコード「０」が記述された場合は、レジスタ番号２で示されるレジスタｒｉに対し直接アクセスする方法である。当該レジスタ番号で示されるレジスタｒｉの値を直接用いることを示している。アクセス方法＃１、＃２に関してコード「１」が記述された場合は、「レジスタ番号１」で示されるレジスタｒｉの値をアドレスとし、当該マイクロプロセッサ１０１で外部メモリ２に対してアクセスする方法である（図６Ｂ参照）。

【００７９】

また、図５Ａ～図５Ｄにおいて、アクセス方法＃１、＃２の後には２ビットの「レジスタ種類１」、「レジスタ種類２」が続けて記述されている。「レジスタ種類１」は「レジスタ番号１」で示されるレジスタｒｉの種類を示し、「レジスタ種類２」は「レジスタ番号２」で示されるレジスタｒｉの種類を示している。レジスタ種類は図６Ｃに示すように２種類が準備されている。「レジスタ種類１」及び「レジスタ種類２」に関して、コード「０」が記述される場合は、レジスタ番号が「３１」以下で使用頻度が高いレジスタｒｉを示している。このレジス

タｒｉ（ｉ＝０～３１）はレジスタ番号をｍ＝５ビットで表現することができる。

【００８０】

この「レジスタ種類１」及び「レジスタ種類２」に関して、「１」が記述される場合は、レジスタ番号が「３２」以上で使用頻度が低いレジスタｒｉを示している。このレジスタｒｉ（ｉ＝３２～８１９１）はレジスタ番号をｎ＝１３ビットで表現するようになされる。このように、レジスタ番号の大きさを区別することでプログラムを圧縮することができる。

【００８１】

この「レジスタ種類１」、「レジスタ種類２」の後には「レジスタ番号１」、「レジスタ番号２」が続けて記述されている。「レジスタ番号１」は例えば、被数を保持するレジスタｒｉを示し、「レジスタ番号２」は加数を保持するレジスタｒｉを示す。

【００８２】

また、図５Ｅに示すｊｕｍｐ命令のフォーマットによれば、最初の２ｂｉｔに命令が記述され、続く２ｂｉｔにはフラグ状態（ｆｌａｇ　ｃｏｎｄｉｔｉｏｎ）が記述される。続く２０ｂｉｔにはジャンプアドレスが記述される。フラグ状態は図６Ｄに示すように、命令実行制御を移すかどうかの判断をするための条件である。コード「０」は「無条件」で常に制御を移す。コード「１」は「ｚｅｒｏ　ｆｌａｇ」でｚｅｒｏ　ｆｌａｇが「１」である場合に、制御を移す。コード「２」は「ｎｏｎ－ｚｅｒｏ　ｆｌａｇ」でｚｅｒｏ　ｆｌａｇが「０」である場合に制御を移すようになされる。コード「３」は未使用である。

【００８３】

続いて、プログラム作成系Ｉにおける処理例について説明をする。図７はプログラム作成系Ｉにおけるプログラム作成例を示す表図である。図７において、Ｐ１はプログラム記述画面をイメージし、Ｐ２にはその記述内容を示し、Ｐ３には実施例で該当する条件を記述したものである。所定のプログラム言語に基づいて目的の演算処理を実行するための命令を、図１に示したプログラム作成装置２００で編集して圧縮プログラムを作成するためである。

【００８４】

プログラム作成装置２００では図１に示した表示装置２４に図７に示すＣ言語によるプログラム記述画面Ｐ１を表示して、キーボード２２及びマウス２３を使用して圧縮プログラムが作成される。このとき、データベース２１からマイクロプロセッサ１０１のプログラム作成に必要なデータが読み出される。例えば、Ｃ言語によるプログラムの記述に必要な「Ｇｌｏｂａｌ変数宣言」、「関数宣言」、「Ｌｏｃａｌ変数宣言」、「代入」、「加算」、「比較」及び「分岐」が読み出される。

【００８５】

この例で新規な設計製造に係るマイクロプロセッサ１０１で、Ｎ＝８１９２個の３２ビットのレジスタｒｉを使用する場合であって、８１９２個のレジスタｒｉに第０番から第８１９１番のシリアル番号を付与したとき、第０番から第３１番のグループのレジスタｒ０～ｒ３１を使用頻度が高い部類として「Ｌｏｃａｌ変数宣言」がなされる。つまり、この例では第３１番目以下のレジスタｒｉに「Ｌｏｃａｌ変数宣言」が割り当てられる。また、第３２番から第８１９１番のグループのレジスタｒ３２～ｒ８１９１を使用頻度が低い部類として「Ｇｌｏｂａｌ変数宣言」がなされる。つまり、第３２番目以上のレジスタｒｉには「Ｇｌｏｂａｌ変数宣言」が割り当てられる。

【００８６】

プログラム作成装置２００では「Ｇｌｏｂａｌ変数宣言」がなされた第３２番から第８１９１番のグループのレジスタｒ３２～ｒ８１９１の命令ビット数をｎ＝１３ビットとしたとき、Ｌｏｃａｌ変数宣言がなされたレジスタｒ０～ｒ３１の命令ビット数は、これよりも８ビット少ないｍ＝５ビットに圧縮される。これと共に、当該プログラムの命令構造の中に「レジスタ種類１」、「レジスタ種類２」を記述して命令長の異なる圧縮プログラムＡＰを作成するようになされる。

【００８７】

図１に示した制御装置２５ではレジスタｒｉの使用頻度に応じて命令の長さを可変するようになされる。この例では、使用頻度が高いレジスタｒ０～ｒ３１を短い命令ビット数ｍ＝５ビットで命令セットし、使用頻度が低いレジスタｒ３２

～ｒ８１９１は長い命令ビット数ｎ＝１３ビットで命令セットするようになされる。このようにすると、頻繁にアクセスするレジスタｒ０～ｒ３１は短い長さの命令をセットすることができ、マイクロプロセッサ１０１に実装されるプログラム格納用のＲＯＭ１４のメモリ容量を削減することができる。

【００８８】

続いて、プログラム作成装置２００におけるコンパイル例について説明をする。図８はプログラム作成装置２００におけるコンパイル例を示すフローチャート（メインルーチン）である。図９はコンパイラにおける代入及び演算処理例を示すフローチャート（サブルーチン）である。

【００８９】

この実施例ではプログラム作成系ＩでＣ言語によるプログラムに基づいて目的の演算処理を実行するための命令を編集して圧縮プログラムを作成する場合を前提とする。また、マイクロプロセッサ１０１が８１９２個のレジスタｒ０～ｒ８１９１を使用する場合であって、８１９２個のレジスタｒｉに第０番から第８１９１番のシリアル番号が付与される場合を例にとる。

【００９０】

これを処理条件にして、プログラム作成系ＩではＣ言語によるプログラムに基づいて命令を編集するために、図８にフローチャートのステップＣ１でプログラムアドレスを「０」にする。その後、ステップＣ２に移行してＣ言語によるプログラムを一行読み込む。このとき、表示装置２４のプログラム記述画面Ｐ１には、例えば、「ｇｌｏｂａｌ変数宣言」を示す

```
int  *read  add  *write  add  , counter, end  val ;
```

が表示され、また、関数宣言を示す

```
void  main   ()
{
```

が表示される。

【００９１】

そして、ステップＣ３で当該プログラムの記述が「ｇｌｏｂａｌ変数宣言」であるかをチェックする。当該記述が「ｇｌｏｂａｌ変数宣言」である場合は、ス

テップＣ４に移行して第３２番目以上のレジスタｒ３２～ｒ８１９１を割り当てる。このグループのレジスタｒ３２～ｒ８１９１を指定する命令ビット数をｍビットとしたとき、ｍ＝１３ビットである。命令は命令形態＃Ｆ１で作成される。その後、ステップＣ１４に移行する。

【００９２】

また、ステップＣ３で当該記述が「ｇｌｏｂａｌ変数宣言」ではない場合はステップＣ５に移行して「ｌｏｃａｌ変数宣言」かをチェックする。このとき、表示装置２４のプログラム記述画面Ｐ１には、例えば、「ｌｏｃａｌ変数宣言」を示す

```
int    temp, added  val ;
```

が表示される。当該記述が「ｌｏｃａｌ変数宣言」である場合は、ステップＣ６に移行して第３１番目以下のレジスタｒ０～ｒ３１を割り当てる。レジスタｒｉを使用する頻度が高いことから、当該レジスタｒｉを指定する命令ビット数ｎを、「ｇｌｏｂａｌ変数宣言」されたレジスタｒ３２～ｒ８１９１よりも８ビット少ない５ビットに圧縮するためである。命令は命令形態＃Ｆ２乃至＃Ｆ４で作成される。その後、ステップＣ１４に移行する。

【００９３】

上述のステップＣ５で当該記述が「ｌｏｃａｌ変数宣言」ではない場合はステップＣ７に移行して、Ｃ言語のプログラムで代入・加算処理等の実行を示す「ｄｏ」が記述されているかをチェックする。このとき、表示装置２４のプログラム記述画面Ｐ１には、例えば、「ｄｏ」を示す

```
do {
        temp  =  *read  add ;
        temp  =  temp  +  added  val ;
        *write  add  =  temp
        read  add  =  read  add  +  added  val ;
        write  add  =  write  add  +  added  val ;
        counter  =  counter  +  added  val ;
      }
```

が表示される。このような代入・加算等の処理を示す「ｄｏ」が記述されている場合は、ステップＣ８に移行して現在のプログラムアドレスを記憶する。その後、ステップＣ１４に移行する。

【００９４】

ステップＣ７で「ｄｏ」が記述されていない場合はステップＣ９に移行してＣ言語のプログラムでその間の処理を示す「ｗｈｉｌｅ」が記述されているかをチェックする。このとき、表示装置２４のプログラム記述画面Ｐ１には、例えば、「ｗｈｉｌｅ」を示す

ｗｈｉｌｅ（counter　!=end　val）；

が表示される。このような比較・分岐等の処理を示す「ｗｈｉｌｅ」が記述されている場合は、ステップＣ１０に移行して代入・演算処理を実行する。

【００９５】

例えば、図９に示すサブルーチンをコールして、そのフローチャートのステップＥ１でＣ言語のプログラムにおいて、当該行が「ｗｈｉｌｅ」を記述している行かがチェックされる。「ｗｈｉｌｅ」が記述されている行の場合は、ステップＥ２に移行して後続の処理で生成する命令をｃｍｐ命令とする。その後、ステップＥ６に移行する。

【００９６】

ステップＥ１で「ｗｈｉｌｅ」が記述されていない行の場合は、ステップＥ３に移行して演算処理は加算かをチェックする。演算処理が加算の場合はステップＥ４に移行して後続の処理で生成する命令をａｄｄ命令とする。演算処理が加算ではない場合はステップＥ５に移行して後続の処理で生成する命令をｌｏａｄ命令とする。その後、ステップＥ６に移行する。

【００９７】

ステップＥ６ではレジスタｒｉに書き込まれる変数に対応するレジスタ番号と、レジスタアレイ１１から読み出される変数に対応するレジスタｒｉのレジスタ番号を調べられる。書込みアドレスＡｗ及び読み出しアドレスＡｒを決めるためである。その後、ステップＥ７に移行する。

【００９８】

ステップＥ７では両方のレジスタ番号が「３２」以上かをチェックする。両方のレジスタ番号が「３２」以上の場合はステップＥ８に移行して図５Ａに示した命令形態＃Ｆ１で命令を生成する。この命令形態＃Ｆ１で「レジスタ種類１」及び「レジスタ種類２」には「１」が記述される。このとき、「レジスタ種類１」及び「レジスタ種類２」は圧縮プログラムの命令構造の中に記述される。例えば、第３２番から第８１９１番のグループのレジスタｒ３２～ｒ８１９１に関して「レジスタ種類１」及び「レジスタ種類２」に「１」が記述される。その後、図８に示したメインルーチンのステップＣ１０にリターンする。

【００９９】

また、ステップＥ７で両方のレジスタ番号が「３２」以上ではない場合はステップＥ９に移行して両方のレジスタ番号が「３１」以下かをチェックする。ここで両方のレジスタ番号が「３１」以下の場合はステップＥ１０に移行して図５Ｄに示した命令形態＃Ｆ４で命令を生成する。この命令形態＃Ｆ４で「レジスタ種類１」及び「レジスタ種類２」には「０」が記述される。このとき、「レジスタ種類１」及び「レジスタ種類２」は圧縮プログラムの命令構造の中に記述される。例えば、第０番から第３１番のグループのレジスタｒ０～ｒ３１に関して「レジスタ種類１」及び「レジスタ種類２」に「０」が記述される。その後、図８に示したメインルーチンのステップＣ１０にリターンする。

【０１００】

更に、両方のレジスタ番号が「３１」以下ではない場合はステップＥ１１に移行してレジスタアレイ１１から読み出される変数のレジスタｒｉの番号が「３２」以上かをチェックする。読み出される変数のレジスタｒｉの番号が「３２」以上の場合は、ステップＥ１２に移行して図５Ｂに示した命令形態＃Ｆ２で命令を生成する。この命令形態＃Ｆ２で「レジスタ種類１」に「０」が記述され、「レジスタ種類２」には「１」が記述される。その後、図８に示したメインルーチンのステップＣ１０にリターンする。

【０１０１】

更にまた、レジスタアレイ１１から読み出される変数のレジスタｒｉの番号が「３２」以上ではない場合は、ステップＥ１３に移行して図５Ｃに示した命令形

態＃Ｆ３で命令を生成する。この命令形態＃Ｆ３では「レジスタ種類１」に「１」が記述され、「レジスタ種類２」に「０」が記述される。その後、図８に示したメインルーチンのステップＣ１０にリターンする。その後、ステップＣ１１に移行してｊｕｍｐ命令を生成する。ｊｕｍｐ命令の飛び先は先に記憶したプログラムアドレスを用いる。その後、ステップＣ１４に移行する。

【０１０２】

上述のステップＣ９で「ｗｈｉｌｅ」が記述されていない場合はステップＣ１２に移行してＣ言語のプログラムにおいて、データの代入又は加算かをチェックする。データの代入又は加算の場合はステップＣ１３に移行してデータの代入又は演算処理を実行する。このステップＣ１３では、図９に示したサブルーチンをコールして、そのフローチャートのステップＥ１～Ｅ１３を経て図８に示したメインルーチンのステップＣ１３にリターンする。その後、ステップＣ１４に移行する。

【０１０３】

また、ステップＣ１２でＣ言語のプログラムにおいて、データの代入又は加算ではない場合はステップＣ１４に移行する。ステップＣ１４ではＣ言語のプログラムに関して最後の行かをチェックされる。最後の行ではない場合は、ステップＣ１５に移行してプログラムアドレスを進める。その後、ステップＣ２に戻って上述したコンパイル処理を繰り返すようになされる。最後の行に至ってこのコンパイル処理を終了する。

【０１０４】

これにより、図５Ａ～図５Ｅに示したような命令形態＃Ｆ１～＃Ｆ５であって、命令長の異なる圧縮プログラムＡＰを作成することができる。この圧縮プログラムＡＰにおいて、第０番から第３１番のグループのレジスタｒ０～ｒ３１を指定する命令ビット数に関してはｍ＝５ビットであり、第３２番から第８１９１番のグループのレジスタｒ３２～ｒ８１９１を指定する命令ビット数に関してはｎ＝１３ビットである。

【０１０５】

続いて、プログラム実行系IIにおける処理例について説明をする。図１０は復

元された演算プログラムによる演算命令の例を示す表図である。図１１はレジスタｒ０，ｒ１・・・・ｒ３２，ｒ３３，ｒ３４，ｒ３５等の状態例、図１２は外部メモリ２におけるデータ格納例を各々示すイメージ図である。

【０１０６】

この例では、データ処理装置１００に接続された外部メモリ２の中に図１２に示すような１０個のメモリセルの配列を二組用意する。一方はメモリ配列＃Ｍ１で、他方はメモリ配列＃Ｍ２である。そして、図１０に示す８つの演算命令（Ｉnstruction）＃Ｉ１～＃Ｉ８に基づいて、その一組のメモリ配列＃Ｍ１に格納された値に「1」を加算し、もう一組のメモリ配列＃Ｍ２にその結果を格納する演算処理の例を挙げる。

【０１０７】

図１０に示す演算命令（Ｉnstruction）＃Ｉ１～＃Ｉ８はＲＯＭ１４の圧縮プログラムＡＰを復元した後の演算プログラムによるものである。この演算プログラムでは図１１に示すように、アクセス頻度が高いレジスタｒｉが二つあるため、これらをそれぞれｒ０とｒ１に割り当てた。これにより、プログラム全体の長さを圧縮する前の演算プログラムに比べて効率よく短縮することができた。

【０１０８】

また、図１１において、レジスタ番号ｒ０で示されるレジスタは一時的に使用され、レジスタ番号ｒ１で示されるレジスタには加算値「1」が格納される。また、レジスタ番号ｒ３２で示されるレジスタには読み出しアドレス「０」が格納され、レジスタ番号ｒ３３で示されるレジスタには書込みアドレス「１０」が格納され、レジスタ番号ｒ３４で示されるレジスタにはカウンタの初期値「０」が格納され、レジスタ番号ｒ３５で示されるレジスタには演算回数値（終了値）「１０」が格納される。

【０１０９】

図１０に示す各々の演算命令＃Ｉ１～＃Ｉ８には、ニーモニックによる表現、機械語による表現及び処理の内容が示されている。演算命令＃Ｉ１は図５に示した命令構造において、機械語で１４００２０ｈによって表されるｌｏａｄ，ｒ０，（ｒ３２）であり、レジスタアレイ１１のレジスタｒ３２の値をアドレスとし

、外部メモリ２から読み出した値をレジスタｒ０に格納する内容である。動作としては例えば、レジスタｒ３２の値を「０」としたとき、図１２に示した外部メモリ２の読み出しアドレスが「０」の内容である、メモリ配列＃Ｍ１のデータ「０」が読み出され、このデータ「０」がレジスタｒ０に書き込まれる。

【０１１０】

演算命令＃Ｉ２は機械語で４００１ｈによって表されるａｄｄ，ｒ０，ｒ１であり、レジスタアレイ１１のレジスタｒ０にレジスタｒ１の値を加算し、その結果をレジスタｒ０に格納する内容である。動作としてはレジスタｒ０の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ０に書き込まれる。

【０１１１】

演算命令＃Ｉ３は機械語で２８０４２０ｈによって表されるｌｏａｄ，（ｒ３３），ｒ０であり、レジスタアレイ１１のレジスタｒ３３の値をアドレスとして、レジスタｒ０の値を外部メモリ２に書き込む内容である。動作としてはレジスタｒ３３が示す外部メモリ２のメモリ配列＃Ｍ２のアドレスにデータ「１」が書き込まれる。

【０１１２】

演算命令＃Ｉ４は機械語で４８０４０１ｈによって表されるａｄｄ，ｒ３２，ｒ１であり、レジスタアレイ１１のレジスタｒ３２にレジスタｒ１の値を加算して、その結果をレジスタｒ３２に格納する内容である。動作としてはレジスタｒ３２の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ３２に書き込まれる。

【０１１３】

演算命令＃Ｉ５は機械語で４８０４２１ｈによって表されるａｄｄ，ｒ３３，ｒ１であり、レジスタアレイ１１のレジスタｒ３３にレジスタｒ１の値を加算して、その結果をレジスタｒ３３に格納する内容である。動作としてはレジスタｒ３３の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ３３に書き込まれる。

【０１１４】

演算命令＃Ｉ６は機械語で４８０４４１ｈによって表されるａｄｄ，ｒ３４，ｒ１であり、レジスタアレイ１１のレジスタｒ３４にレジスタｒ１の値を加算して、その結果をレジスタｒ３４に格納する内容である。動作としてはレジスタｒ３４の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ３４に書き込まれる。この演算命令＃Ｉ４～＃Ｉ６によって実行ステートマシーン５１内のカウンタでは読み出しアドレスＡｒ及び書き込みアドレスＡｗに関して「１」が加算される。

【０１１５】

演算命令＃Ｉ７は機械語で８Ｃ０４４０２３ｈによって表されるｃｍｐ，ｒ３４，ｒ３５であり、レジスタアレイ１１のレジスタｒ３４の内容とレジスタｒ３５の内容とを比較し、その値が同じ場合は、ｚｅｒｏ　ｆｌａｇに「１」をセットし、異なっている場合は「０」にセットする内容である。動作としてはレジスタｒ３４とレジスタｒ３５の値である「１」と「１０」は異なるので、ｚｅｒｏ　ｆｌａｇには「０」がセットされる。ｚｅｒｏ　ｆｌａｇの値はラッチ回路５１０によって保持され、以降の命令によって参照される。

【０１１６】

演算命令＃Ｉ８は機械語でＥ００００００ｈによって表されるｊｕｍｐ　ｎｚ，ＬＯＯＰであり、ｚｅｒｏ　ｆｌａｇが「０」の場合は、ＬＯＯＰで示されるラベルへ制御を移す内容である。動作としては、ｚｅｒｏ　ｆｌａｇが「０」の場合は制御を演算命令＃Ｉ１に移す。上記の動作が１０回、繰り返されるとレジスタｒ３４の値が「１０」になり、演算命令＃Ｉ７によりｚｅｒｏ　ｆｌａｇが「１」にセットされ、演算命令＃Ｉ８で制御が移らなくなり、演算処理を終了するようになされる。このように、全てのレジスタｒｉの命令ビット数を単一の方法で表現した場合と比べて、効率良くＲＯＭ１４を使用することが可能になる。

【０１１７】

続いて、マイクロプロセッサ１０１における動作例について説明をする。図１３はマイクロプロセッサ１０１における動作例を示すフローチャートである。図１４及び図１５は命令ビット復元デコーダ１３における処理例（その１，２）を示すフローチャートである。

【０１１８】

この実施例ではマイクロプロセッサ１０１がプログラム実行系IIを構成し、ＲＯＭ１４から読み出された圧縮プログラムＡＰから図１０に示した演算命令＃Ｉ１～＃Ｉ８を含む演算プログラムを復元する。このとき、命令形態＃Ｆ１～＃Ｆ４に関して「レジスタ種類１」及び、「レジスタ種類２」にコード「０」が記述されている場合は、命令ビット復元デコーダ１３によってレジスタ番号の拡張が行われる。この際の命令ビットの拡張では、例えば、「レジスタ番号１」の命令ビット数ｍ＝５ビットの上位に８ビットの「０」を追加するようになされる。この演算プログラムに基づいて、図１２に示した外部メモリ２の中のメモリ配列＃Ｍ１の値に「１」を加算し、メモリ配列＃Ｍ２に格納するようになされる。

【０１１９】

レジスタアレイ１１のレジスタ状態については、図１１に示したように例えば、６個のレジスタｒ０，ｒ１，ｒ３２，ｒ３３，ｒ３４，ｒ３５に関して、ｒ０が不定、ｒ１が初期値「１」、ｒ３２及びｒ３４が共に初期値「０」、ｒ３３及びｒ３５が初期値「１０」が設定される。これらの値を書き込む場合は、実行ステートマシーン５１ではレジスタアレイ１１に書き込みアドレスＡｗが出力され、その初期値「０」、「１」、「１０」が設定される。

【０１２０】

これを動作条件にして、図１３に示すフローチャートのステップＦ１で、まず、命令ビット復元デコーダ１３はＲＯＭ１４から圧縮プログラム（機械語命令）ＡＰを順次受け取り、このプログラムＡＰを解読して所定の命令長の演算命令＃Ｉ１～＃Ｉ８を検出する。

【０１２１】

この命令ビット復元デコーダ１３は例えば、図１４に示すサブルーチンをコールしてそのフローチャートのステップＧ１で命令部分を取り出し、命令信号Ｓ９を実行ステートマシーン５１に出力する。これと共に、命令ビット復元デコーダ１３ではステップＧ２に移行して当該命令形態がｊｕｍｐ命令であるかをチェックする。当該命令形態が＃Ｆ５で示されるｊｕｍｐ命令の場合はステップＧ１２に移行してｆｌａｇ　ｃｏｎｄｉｔｉｏｎ、ｊｕｍｐ　ａｄｄｒｅｓｓを出力す

る。その後、メインルーチンのステップＦ１にリターンする。

【０１２２】

また、ステップＧ２で当該命令形態がｊｕｍｐ命令でない場合は、ステップＧ４に移行して当該命令形態に関して「レジスタ種類１」に記述されているコードは「０」又は「１」かをチェックする。「レジスタ種類１」にコード「０」が記述されている場合はステップＧ４に移行して「レジスタ番号１」の命令ビット数ｍを５ｂｉｔ長として圧縮プログラムＡＰから取り出す。その後、ステップＧ５に移行して「レジスタ番号１」の命令ビット数ｍ＝５ｂｉｔの上位８ビットに「０」を付加して１３ｂｉｔ長とする。その後、ステップＧ７に移行する。

【０１２３】

上述のステップＧ３で「レジスタ種類１」にコード「１」が記述されている場合はステップＧ６に移行して「レジスタ番号１」の命令ビット数ｎを１３ｂｉｔ長として圧縮プログラムＡＰから取り出す。その後、ステップＧ７に移行して、当該命令形態に関して「レジスタ種類２」に記述されているコードは「０」又は「１」かをチェックする。「レジスタ種類２」にコード「０」が記述されている場合はステップＧ８に移行して「レジスタ番号２」の命令ビット数ｍを５ｂｉｔ長として圧縮プログラムＡＰから取り出す。その後、ステップＧ９に移行して「レジスタ番号２」の命令ビット数ｍ＝５ｂｉｔの上位８ビットに「０」を付加して１３ｂｉｔ長とする。その後、ステップＧ１１に移行する。

【０１２４】

上述のステップＧ７で「レジスタ種類２」にコード「１」が記述されている場合はステップＧ１０に移行して「レジスタ番号２」の命令ビット数ｎを１３ｂｉｔ長として圧縮プログラムＡＰから取り出す。その後、ステップＧ１１に移行して、「レジスタ番号１」、「レジスタ番号２」、「アクセス方法＃１」及び「アクセス方法＃２」を検出する。

【０１２５】

その後、図１３に示したメインルーチンのステップＦ１にリターンする。従って、実行ステートマシン５１には「レジスタ種類１」及び「レジスタ種類２」は出力されず、所定の命令長の演算命令＃Ｉ１～＃Ｉ８に基づく命令制御信号Ｓ４

、命令信号Ｓ９及び各引数信号Ｓ１０が出力される。

【０１２６】

この命令信号Ｓ９にはｌｏａｄ命令、ａｄｄ命令、ｃｍｐ命令、ｊｕｍｐ命令が含まれる。各引数信号Ｓ１０にはアクセス方法＃１，アクセス方法＃２，レジスタ番号ｒ０，ｒ１・・・等、フラグ状態（ｆｌａｇ　ｃｏｎｄｉｔｉｏｎ）及びジャンプアドレス等が含まれる。命令制御信号Ｓ４はデコーダ１３から命令読出しステートマシーン５２に出力される。

【０１２７】

なお、ＲＯＭ１４で圧縮プログラムＡＰを読み出す場所（アドレス）はプログラムカウンタ５４（ＰＣ）によって指定される。これらの読み出しの動作は命令読み出しステートマシーン５２によって制御される。

【０１２８】

命令読出しステートマシーン５２では命令ビット復元デコーダ１３から出力される命令制御信号Ｓ４に基づいてプログラムカウンタ５４及び実行ステートマシーン５１を制御する。例えば、当該マシーン５２は命令ビット復元デコーダ１３から命令信号Ｓ９及び各引数信号Ｓ１０が実行ステートマシーン５１へ出力されると共に命令実行開始信号Ｓ２９を出力する。

【０１２９】

実行ステートマシーン５１では命令実行開始信号Ｓ２９に基づいて命令の実行を開始する。例えば、データの書込み時には、書込み制御信号Ｓｗがレジスタアレイ１１に出力され、セレクタ５９には選択制御信号Ｓ２４が出力される。データの読出し時には、レジスタアレイ１１に読出しアドレスＡｒが出力される。

【０１３０】

演算時には、ラッチ制御信号Ｓ３４がラッチ回路５８に出力され、ラッチ回路５１０にはラッチ制御信号Ｓ３８が出力される。当該プロセッサ外部には外部制御信号Ｓ１６が出力される。命令の実行が終了すると、実行ステートマシーン５１は命令読み出しステートマシーン５２へ実行終了信号Ｓ２６を出力し、プログラムカウンタ５４の値を進めるようになされる。

【０１３１】

プログラムカウンタ５４ではカウント制御信号Ｓ３０に基づいてＲＯＭ１４から圧縮プログラムＡＰを読み出す場所が指定される。＋１インクリメンタ５５はプログラムカウンタ５４のカウント出力信号Ｓ５を「＋１」してインクリメンとするようになされる。

【０１３２】

そして、ステップＦ２で実行ステートマシーン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ１を受け取ると、機械語で１４００２０ｈによって表されるｌｏａｄ，ｒ０，（ｒ３２）に基づいて書込み信号Ｓ１６を外部メモリ２に出力する。この値はレジスタアレイ１１の読み出しアドレスＡｒとして用いられる。レジスタアレイ１１は３２番目の値をデータ信号線Ｌ２０に出力する。

【０１３３】

この値はラッチ回路５１１によって保持され、アドレスバス１９Ｂを経由し、外部メモリ２へ出力される。そして、レジスタアレイ１１のレジスタｒ３２の値をアドレスとし、外部メモリ２から読み出した値をレジスタｒ０に格納する。このとき、レジスタｒ３２の値が「０」であるので、図１２に示した外部メモリ２のアドレスが「０」の内容であるメモリ配列＃Ｍ１のデータ「０」が読み出され、このデータ「０」がレジスタｒ０に書き込まれる。

【０１３４】

つまり、外部メモリ２からアドレスバス１９Ｂによって転送されたアドレス（場所）のデータがセレクタ５９に出力される。実行ステートマシーン５１ではこのデータが選択されるように、選択信号Ｓ２４を出力する。これにより、データがレジスタアレイ１１に入力される。そして、実行ステートマシーン５１では書き込みアドレスＡｗ＝「０」をレジスタアレイ１１に出力する。その後、実行ステートマシン５１は書き込み信号Ｓｗを用いて実際に、演算結果値の書き込みを指示するようになされる。

【０１３５】

その後、ステップＦ３で実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ２を受け取ると、機械語で４００

１ｈによって表されるａｄｄ，ｒ０，ｒ１に基づいてレジスタアレイ１１のレジスタｒ０にレジスタｒ１の値を加算し、その結果をレジスタｒ０に格納する。このとき、レジスタｒ０の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ０に書き込まれる。

【０１３６】

そして、ステップＦ４で実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ３を受け取ると、機械語で２８０４２０ｈによって表されるｌｏａｄ，（ｒ３３），ｒ０に基づいてレジスタアレイ１１のレジスタｒ３３の値をアドレスとして、レジスタｒ０の値を外部メモリ２に書き込む。このとき、レジスタｒ３３が示す外部メモリ２のメモリ配列＃Ｍ２のアドレスにデータ「１」が書き込まれる。

【０１３７】

その後、ステップＦ５で実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ４を受け取ると、機械語で４８０４０１ｈによって表されるａｄｄ，ｒ３２，ｒ１に基づいてレジスタアレイ１１のレジスタｒ３２にレジスタｒ１の値を加算して、その結果をレジスタｒ３２に格納する。このとき、レジスタｒ３２の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ３２に書き込まれる。この演算命令＃Ｉ４によって実行ステートマシーン５１内のカウンタでは読み出しアドレスＡｒ及び書き込みアドレスＡｗに関して「１」が加算される。

【０１３８】

そして、ステップＦ６で実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ５を受け取ると、機械語で４８０４２１ｈによって表されるａｄｄ，ｒ３３，ｒ１に基づいてレジスタアレイ１１のレジスタｒ３３にレジスタｒ１の値を加算して、その結果をレジスタｒ３３に格納する。このとき、レジスタｒ３３の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ３３に書き込まれる。この演算命令＃Ｉ５によって実行ステートマシーン５１内のカウンタでは読み出しアドレスＡｒ及び書き込みアドレスＡｗに関して「１」が加算される。

【０１３９】

その後、ステップＦ７で実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ６を受け取ると、機械語で４８０４４１ｈによって表されるａｄｄ，ｒ３４，ｒ１に基づいてレジスタアレイ１１のレジスタｒ３４にレジスタｒ１の値を加算して、その結果をレジスタｒ３４に格納する。このとき、レジスタｒ３４の内容である「０」にレジスタｒ１の値である「１」が加算され、その結果「１」がレジスタｒ３４に書き込まれる。この演算命令＃Ｉ６によって実行ステートマシーン５１内のカウンタでは読み出しアドレスＡｒ及び書き込みアドレスＡｗに関して「１」が加算される。

【０１４０】

この例では、ステップＦ８でレジスタｒ３５が示す値＝１０回に至ったかが判別される。例えば、実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ７を受け取ると、機械語で８Ｃ０４４０２３ｈによって表されるｃｍｐ，ｒ３４，ｒ３５に基づいてレジスタアレイ１１のレジスタｒ３４の内容とレジスタｒ３５の内容とを比較し、その値が同じ場合は、ｚｅｒｏ　ｆｌａｇに「１」をセットし、異なっている場合は「０」にセットする。このとき、レジスタｒ３４とレジスタｒ３５の値である「１」と「１０」は異なるので、ｚｅｒｏ　ｆｌａｇには「０」がセットされる。ｚｅｒｏ　ｆｌａｇの値はラッチ回路５１０によって保持され、以降の命令によって参照される。

【０１４１】

そして、実行ステートマシン５１は命令読出しステートマシーン５２の命令読出し制御を受けて演算命令＃Ｉ８を受け取ると、機械語でＥ００００００ｈによって表されるｊｕｍｐ　ｎｚ，ＬＯＯＰに基づいてｚｅｒｏ　ｆｌａｇが「０」の場合は、ＬＯＯＰで示されるラベルへ制御を移行する。ｚｅｒｏ　ｆｌａｇが「０」の場合は制御をステップＦ２の演算命令＃Ｉ１に移す。上記の動作がステップＦ８で１０回繰り返されるとレジスタｒ３４の値が「１０」になり、演算命令＃Ｉ７によりｚｅｒｏ　ｆｌａｇが「１」にセットされ、演算命令＃Ｉ８で制御が移らなくなり、演算処理を終了するようになされる。

【０１４２】

このように、本発明に係る実施例としてのマイクロプロセッサ１０１によれば、８１９２個のレジスタｒ０～ｒ８１９１の中でその使用頻度に基づいて当該レジスタｒｉを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中に「レジスタ種類１」及び「レジスタ種類２」が記述された命令長の異なる圧縮プログラムＡＰに基づいてデータを処理するようになされる。

【０１４３】

ＲＯＭ１４には８１９２個のレジスタｒ０～ｒ８１９１の中から当該レジスタｒｉを指定するための圧縮プログラムＡＰが記憶される。命令ビット復元デコーダ１３では、このＲＯＭ１４から圧縮プログラムＡＰを読み出して「レジスタ種類１」及び「レジスタ種類２」が解読され、この「レジスタ種類１」及び「レジスタ種類２」に基づいて当該レジスタｒｉを指定するための命令ビット数が復元される。

【０１４４】

従って、レジスタｒｉの使用頻度に応じて可変された命令の長さの圧縮プログラムＡＰであって、頻繁にアクセスするレジスタｒ０～ｒ３１には短い長さの命令がセットされた、圧縮プログラムデータをＲＯＭ１４に格納することができるので、そのメモリ容量を低減することができる。この例ではプログラムを圧縮する前に比べて１６ビット×３２個×命令数分だけメモリ容量を低減できる。

【０１４５】

これにより、メモリセルや論理演算素子から成るＰＬＤによりマイクロプロセッサ１０１を構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができるようになる。

【０１４６】

【発明の効果】

以上説明したように、本発明に係るデータ処理システムによれば、一方で、所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデ

ータを処理する場合に、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するデータ処理装置を備えるものである。

【０１４７】

この構成によって、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。従って、プログラム実行系ではプログラムデータを格納するＲＯＭ等のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

【０１４８】

本発明に係るデータ処理装置によれば、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムに基づいてデータを処理する場合に、この圧縮プログラムから解読したレジスタ種類に基づいて当該レジスタを指定するための命令ビット数を復元する命令解読復元手段を備え、この命令解読復元手段によって復元された所定長の命令に基づいてレジスタを指定して任意の演算を実行するものである。

【０１４９】

この構成によって、レジスタの使用頻度に応じて可変された命令の長さの圧縮プログラムであって、頻繁にアクセスするレジスタには短い長さの命令がセットされた、圧縮プログラムデータを記憶手段に格納することができるので、そのメモリ容量を低減することができる。従って、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

【０１５０】

本発明に係るデータ処理方法によれば、プログラム作成系で所定のプログラム言語に基づいて目的の演算処理を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用してデータを処理する場合に、プログラム作成系では、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成し、プログラム実行系では、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、ここで解読されたレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、ここで復元された所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するようになされる。

【０１５１】

この構成によって、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。従って、プログラム実行系ではプログラムデータを格納するＲＯＭ等のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るＰＬＤによりプロセッサを構築する場合に、ＲＯＭとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

【０１５２】

この発明は命令実行プログラム等に基づいて各種データ処理をするＣＰＵや、ＭＰＵ、ＰＬＤ等、これらの組み込み電子機器に適用して極めて好適である。

【図面の簡単な説明】

【図１】

本発明に係る実施形態としてのデータ処理システム１０の構成例を示すブロック図である。

【図２】

データ処理システム１０における処理例を示すフローチャートである。

【図３】

本発明に係る実施例としてのマイクロプロセッサ１０１の構成例を示すブロッ

ク図である。

【図４】

レジスタアレイ１１の内部構成例を示すブロック図である。

【図５】

Ａ～ＥはＲＯＭ１４にセットされる命令の構造例を示すフォーマットである。

【図６】

Ａ～Ｄは命令構造における記述内容例を示す表図である。

【図７】

プログラム作成系Ｉにおけるプログラム作成例を示す表図である。

【図８】

プログラム作成装置２００におけるコンパイル例を示すフローチャート（メインルーチン）である。

【図９】

コンパイラにおける代入及び演算処理例を示すフローチャート（サブルーチン）である。

【図１０】

復元された演算プログラムによる演算命令の例を示す表図である。

【図１１】

ｒ０，ｒ１・・・ｒ３２，ｒ３３，ｒ３４，ｒ３５等のレジスタの状態例を示すイメージ図である。

【図１２】

外部メモリ２におけるデータ格納例を示すイメージ図である。

【図１３】

マイクロプロセッサ１０１における動作例を示すフローチャートである。

【図１４】

命令ビット復元デコーダ１３における処理例（その１）を示すフローチャートである。

【図１５】

命令ビット復元デコーダ１３における処理例（その２）を示すフローチャート

である。

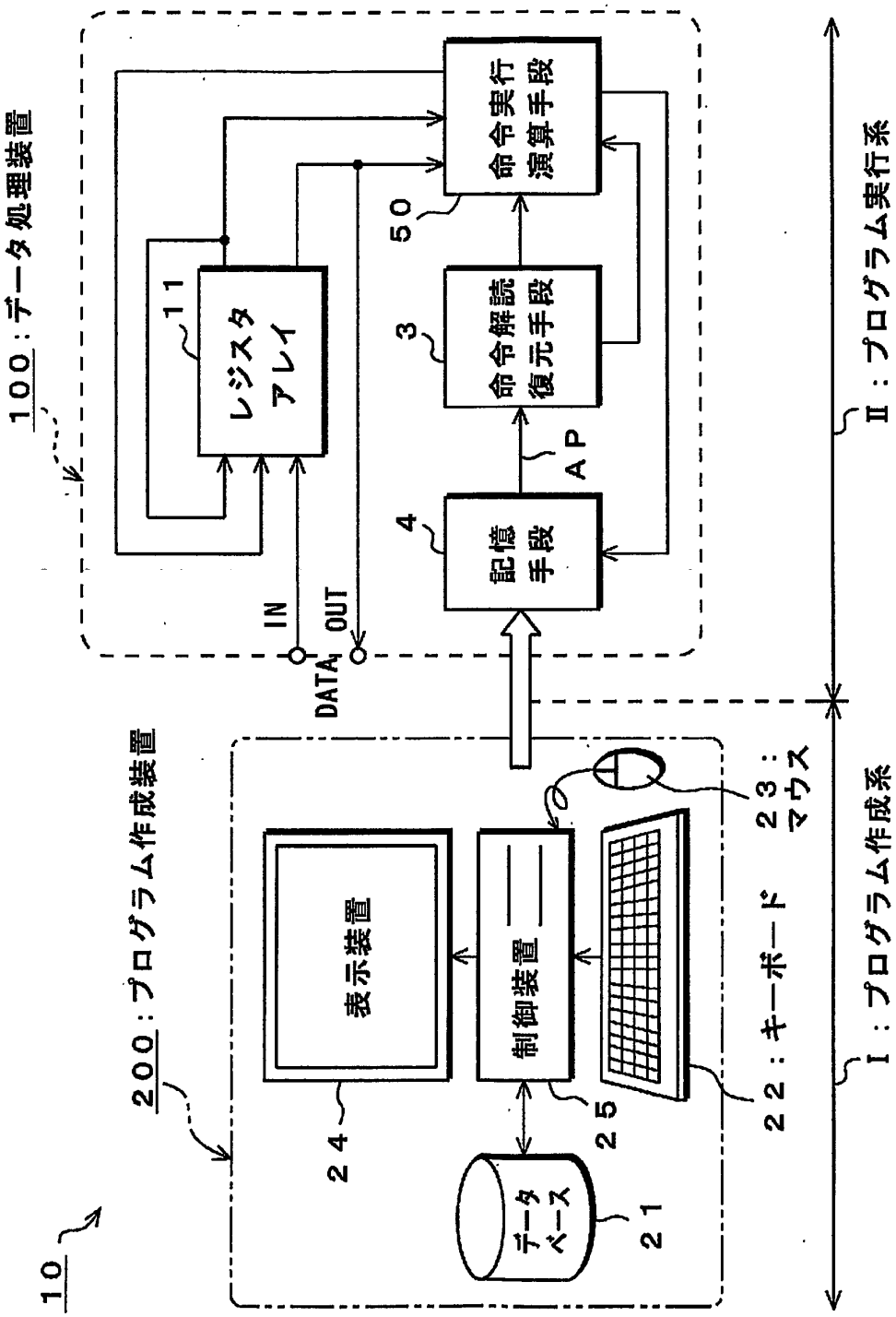
【符号の説明】

２・・・外部メモリ、３・・・命令解読復元手段、４・・・記憶手段、１０・・・データ処理システム、１１・・・レジスタアレイ、１２・・・ＡＬＵ、１３・・・命令ビット復元デコーダ（命令解読復元手段）、１４・・・ＲＯＭ（記憶手段）、５０・・・命令実行演算手段、１００・・・データ処理装置、１０１・・・マイクロプロセッサ、２００・・・プログラム作成装置

【書類名】　図面

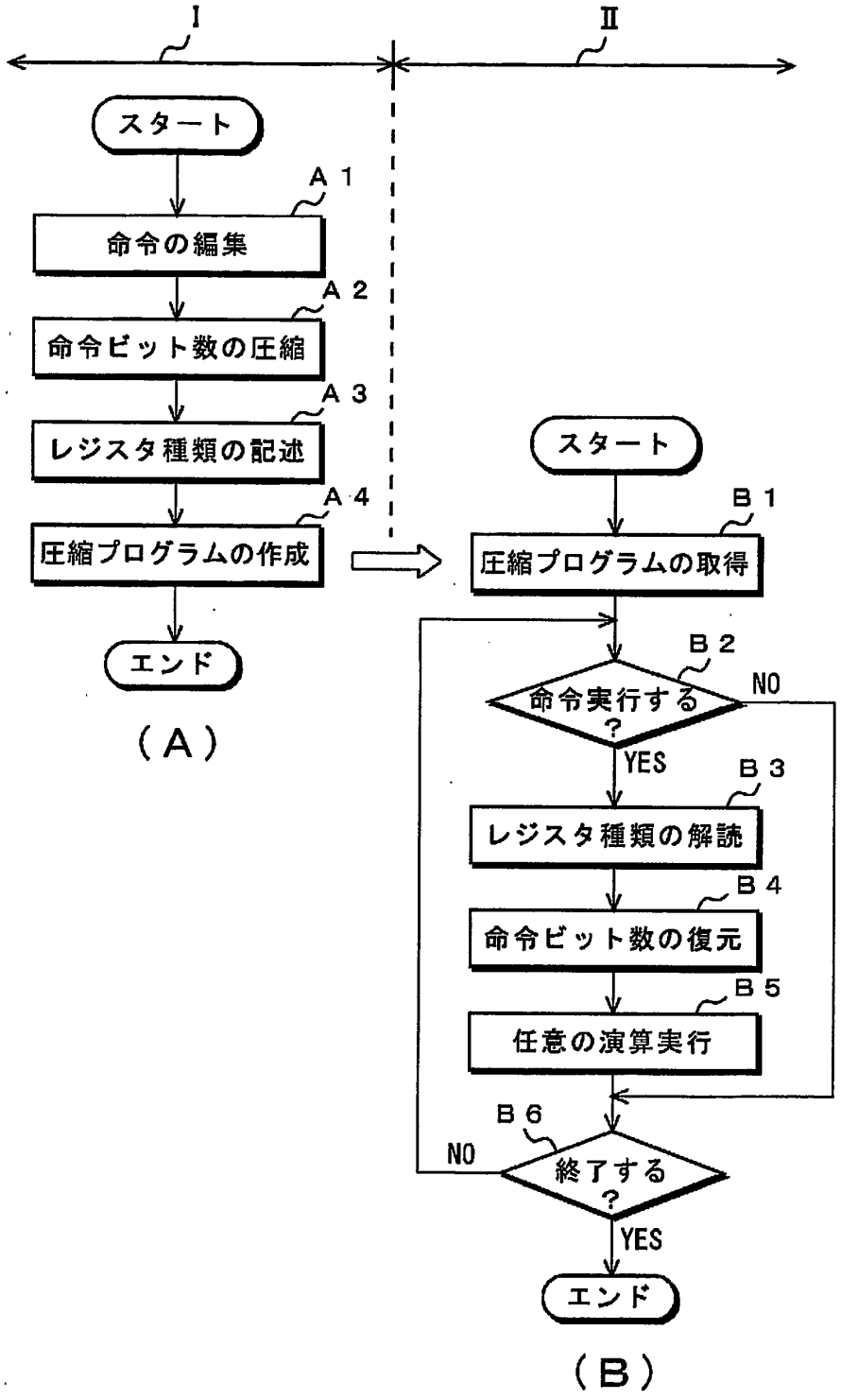
【図１】

実施形態としてのデータ処理システム１０の構成例



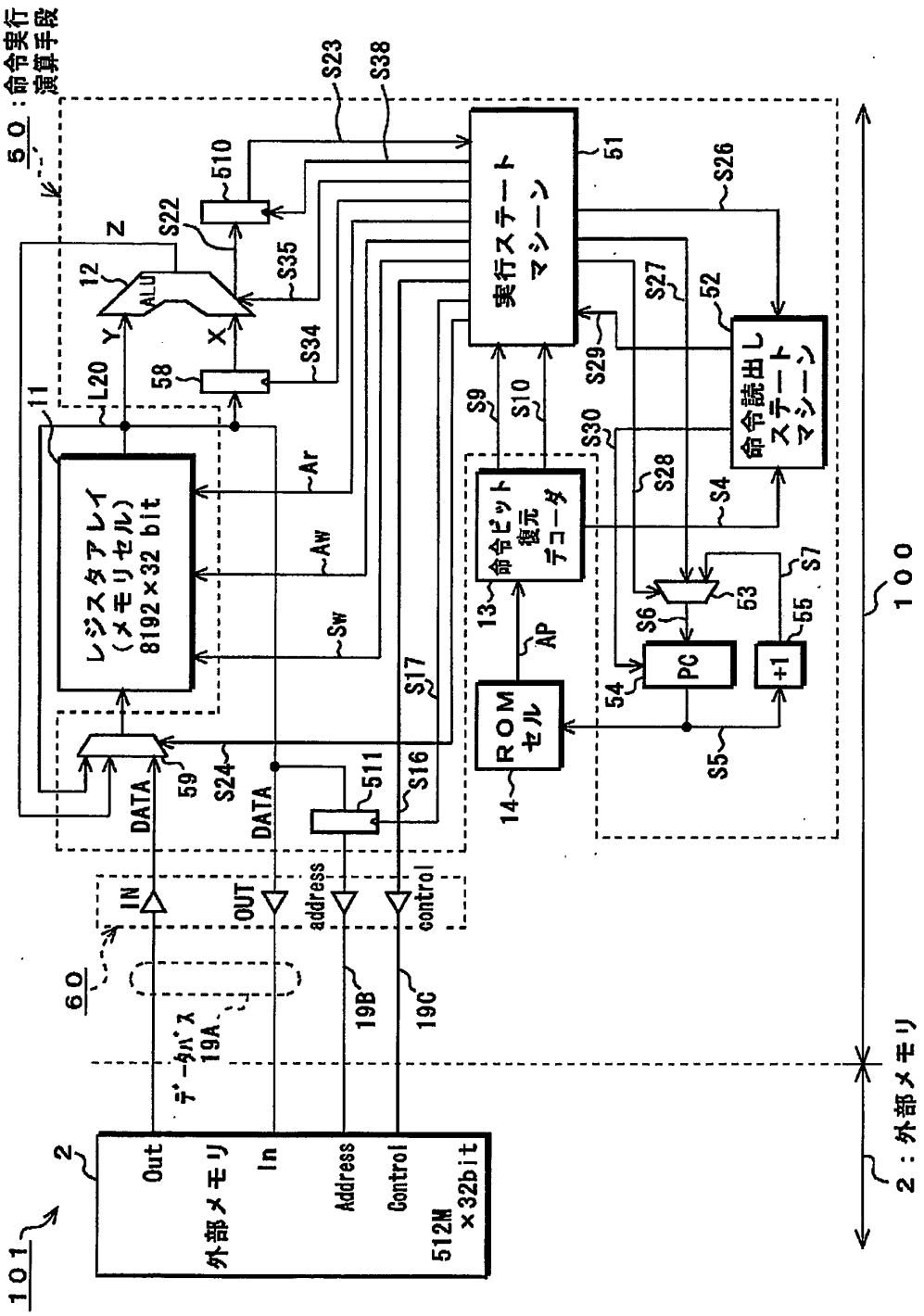
【図２】

データ処理システム１０における処理例



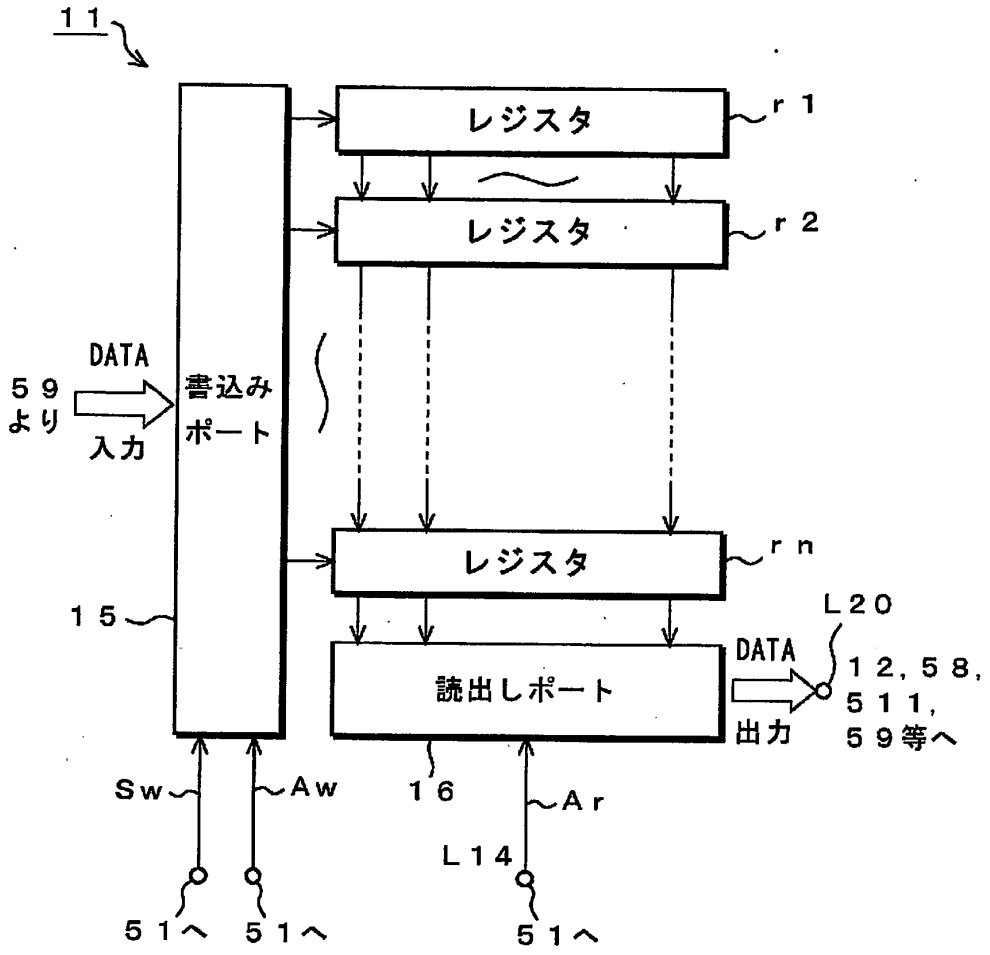
【図３】

実施例としてのマイクロプロセッサ１０１の構成例



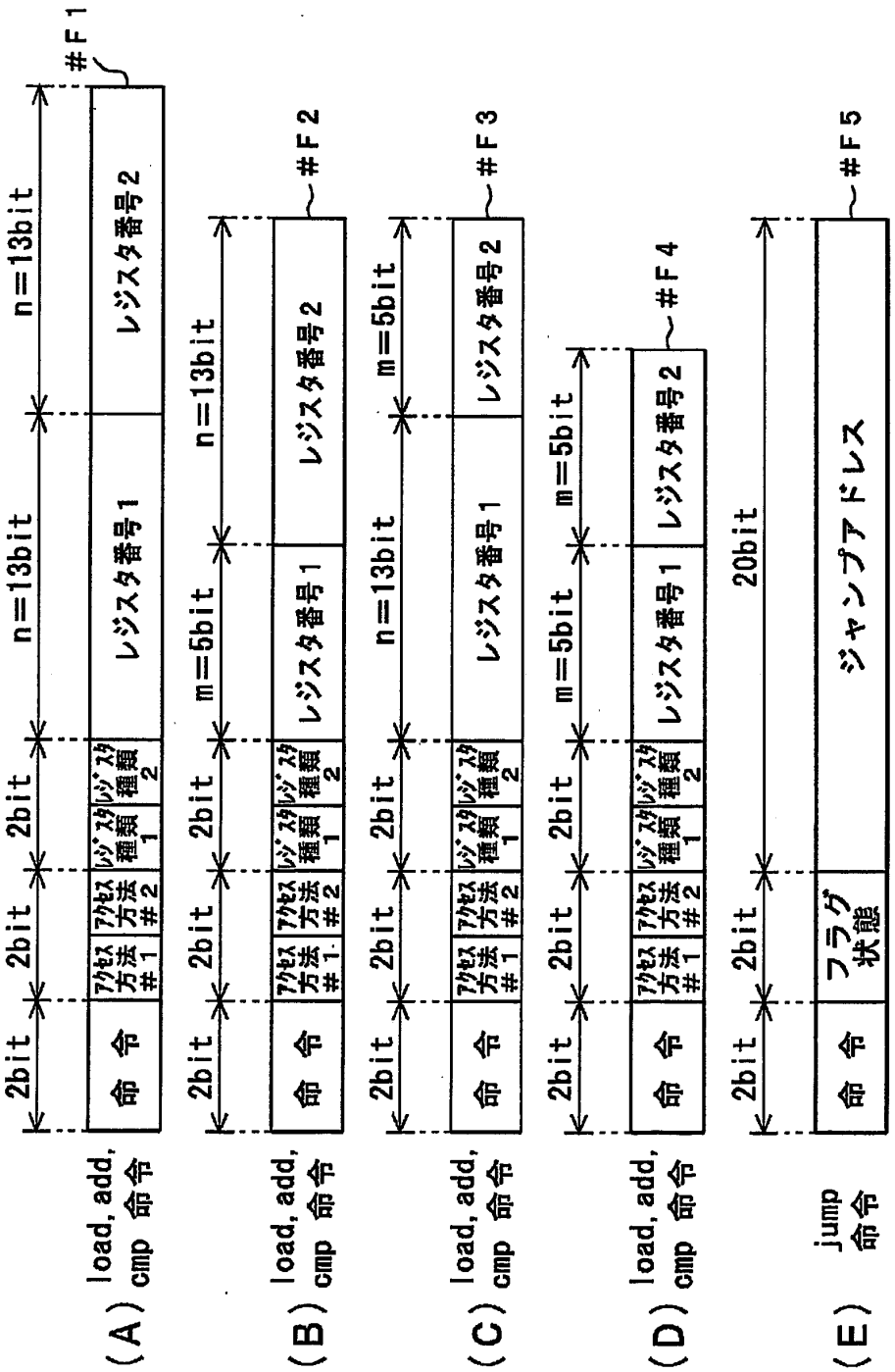
【図４】

レジスタアレイ１１の内部構成例



【図５】

ＲＯＭ１４にセットされる命令の構造例



【図６】

命令構造における記述内容例

命令	
0	load
1	add
2	cmp
3	jump

（Ａ）

アクセス方法＃１，＃２	
0	レジスタ直接
1	レジスタ相対

（Ｂ）

レジスタ種類１，２	
0	レジスタ番号が３１以下
1	レジスタ番号が３２以上

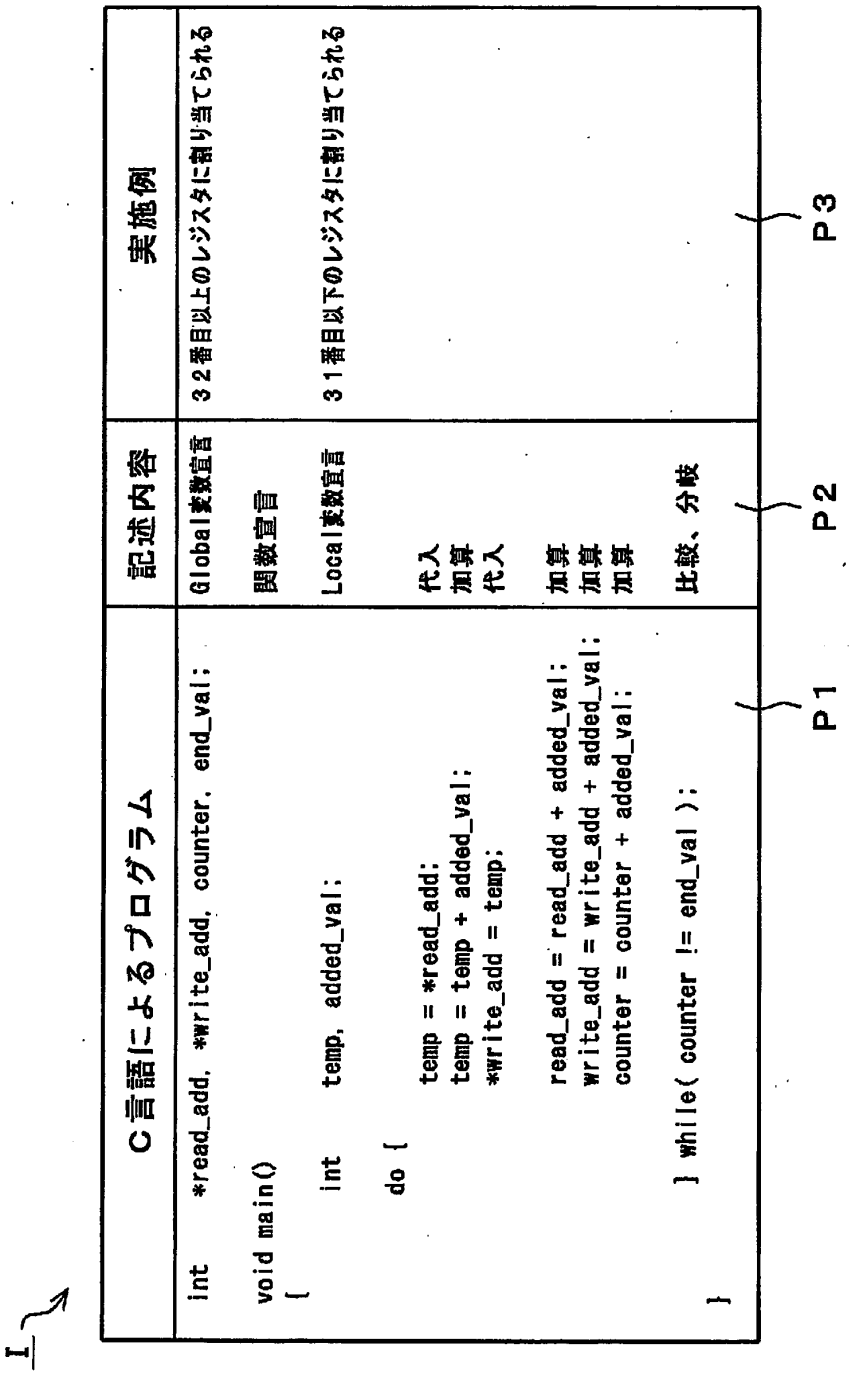
（Ｃ）

フラグ状態	
0	無条件
1	zero flag
2	non-zero flag
3	未使用

（Ｄ）

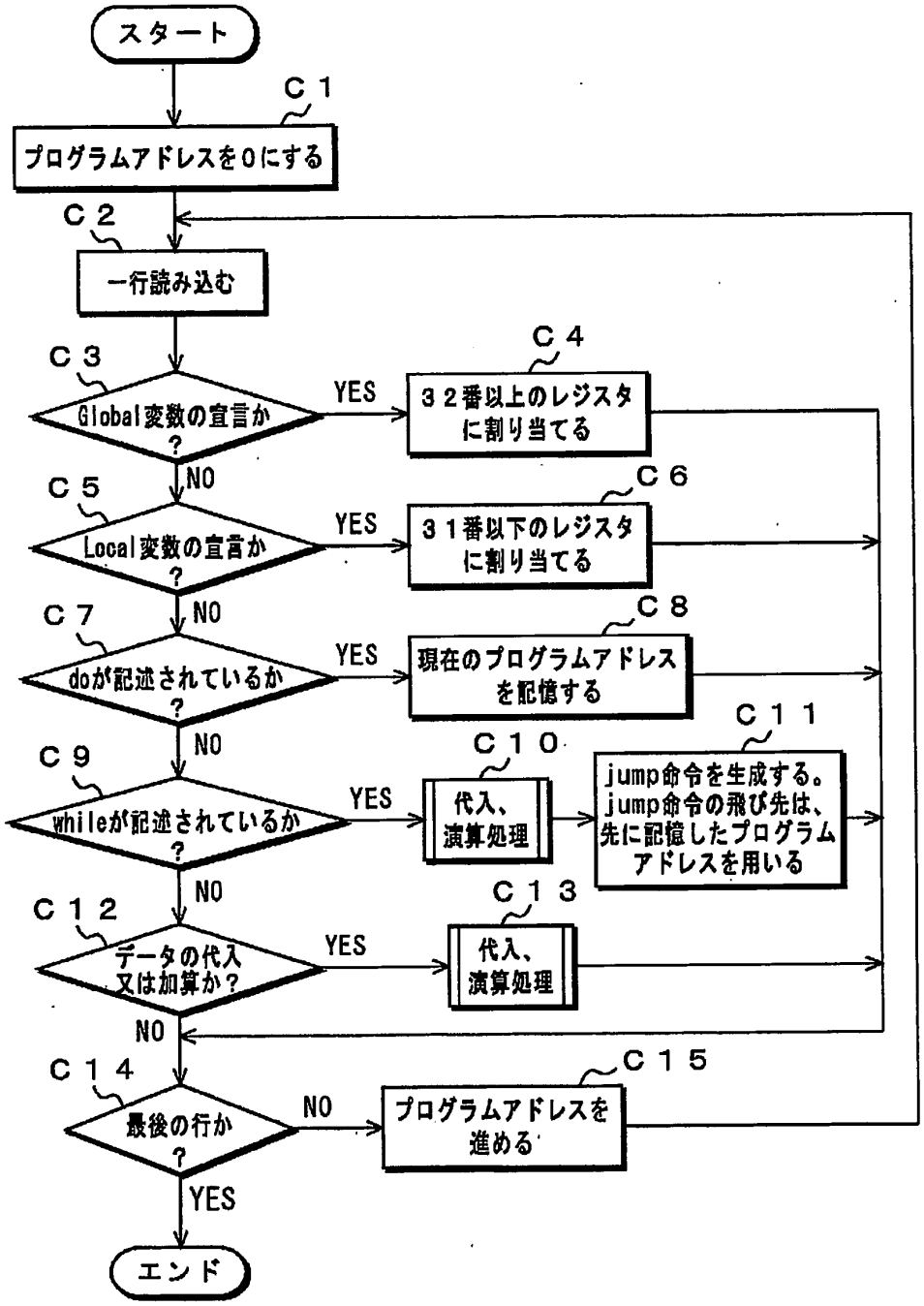
【図７】

プログラム作成系Ｉにおけるプログラム作成例



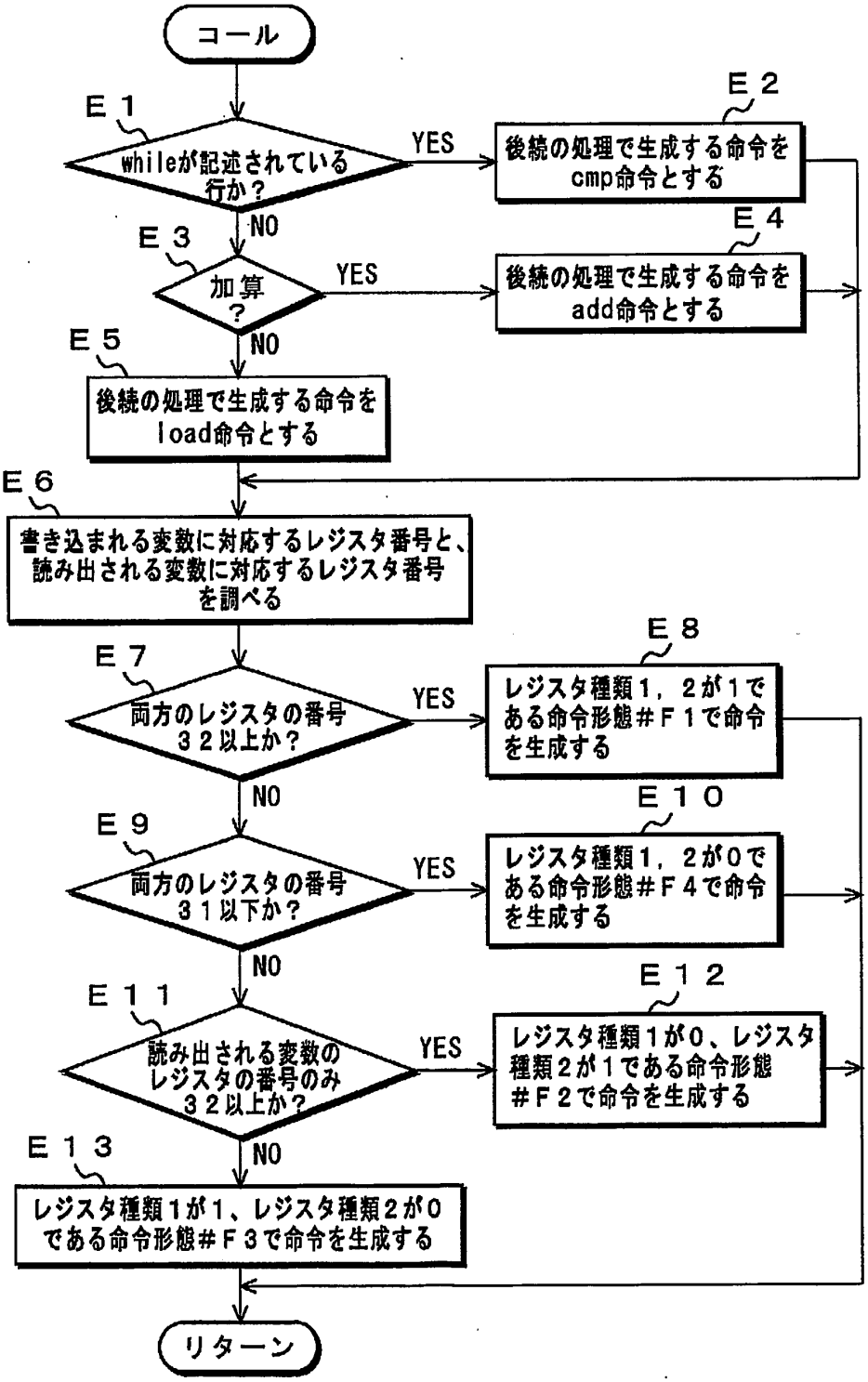
【図８】

プログラム作成装置２００におけるコンパイル例



【図９】

代入及び演算処理例



【図１０】

演算プログラムによる演算命令の例

Ｎｏ.	ニーモニックによる表現		機械語による表現
＃Ｉ１	LOOP	load r0,(r32)	140020h
＃Ｉ２		add r0, r1	4001h
＃Ｉ３		load(r33), r0	280420h
＃Ｉ４		add r32, r1	480401h
＃Ｉ５		add r33, r1	480421h
＃Ｉ６		add r34, r1	480441h
＃Ｉ７		cmp r34, r35	8C044023h
＃Ｉ８		jump nz, LOOP	E00000h

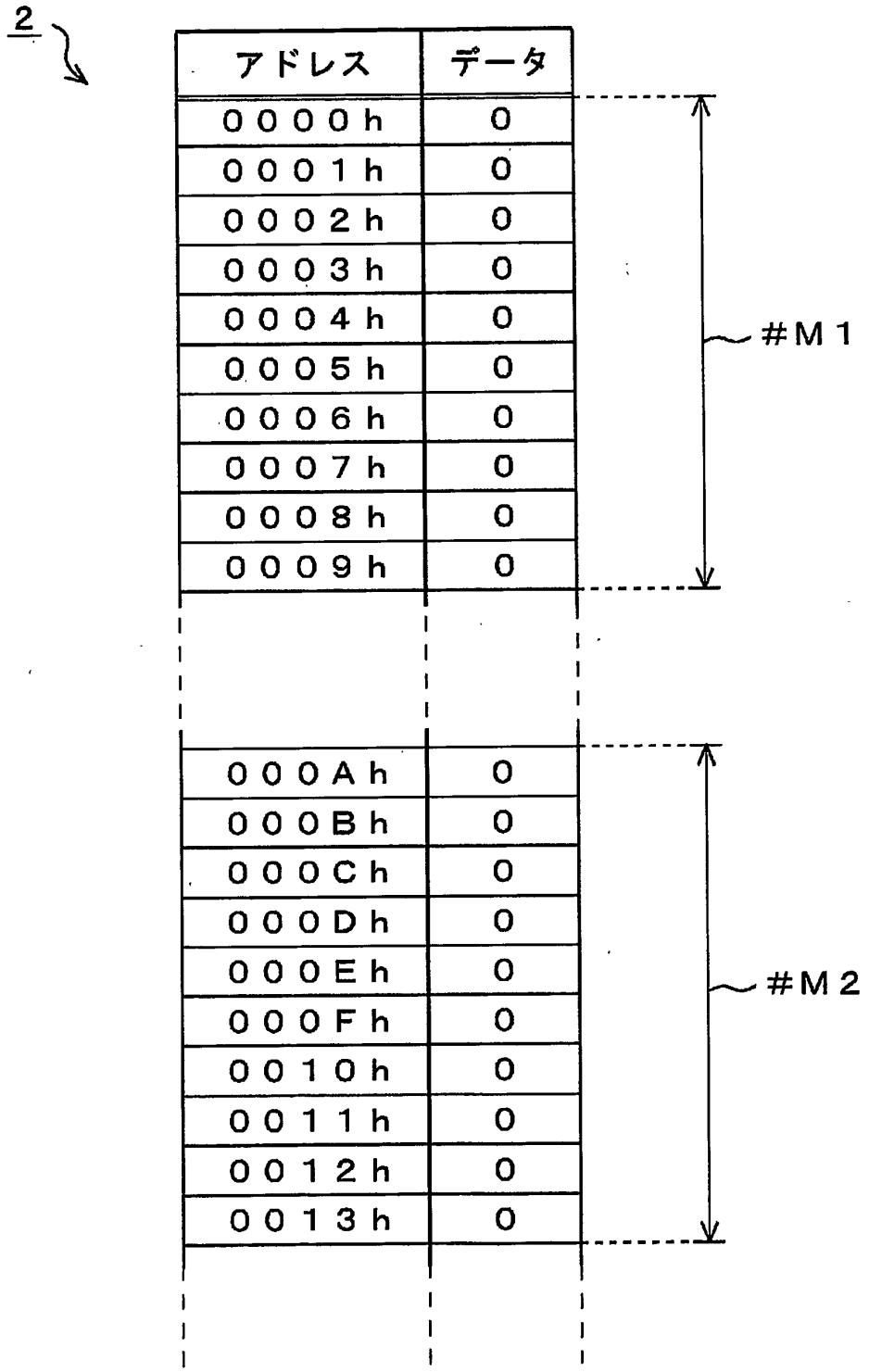
【図１１】

レジスタの状態例

レジスタ番号	値	
ｒ０	—	← 一時的に使用されるレジスタ
ｒ１	１	← 加算値
ｒ３２	０	← 読み出しアドレス
ｒ３３	１０	← 書き込みアドレス
ｒ３４	０	← カウンタ
ｒ３５	１０	← 終了値

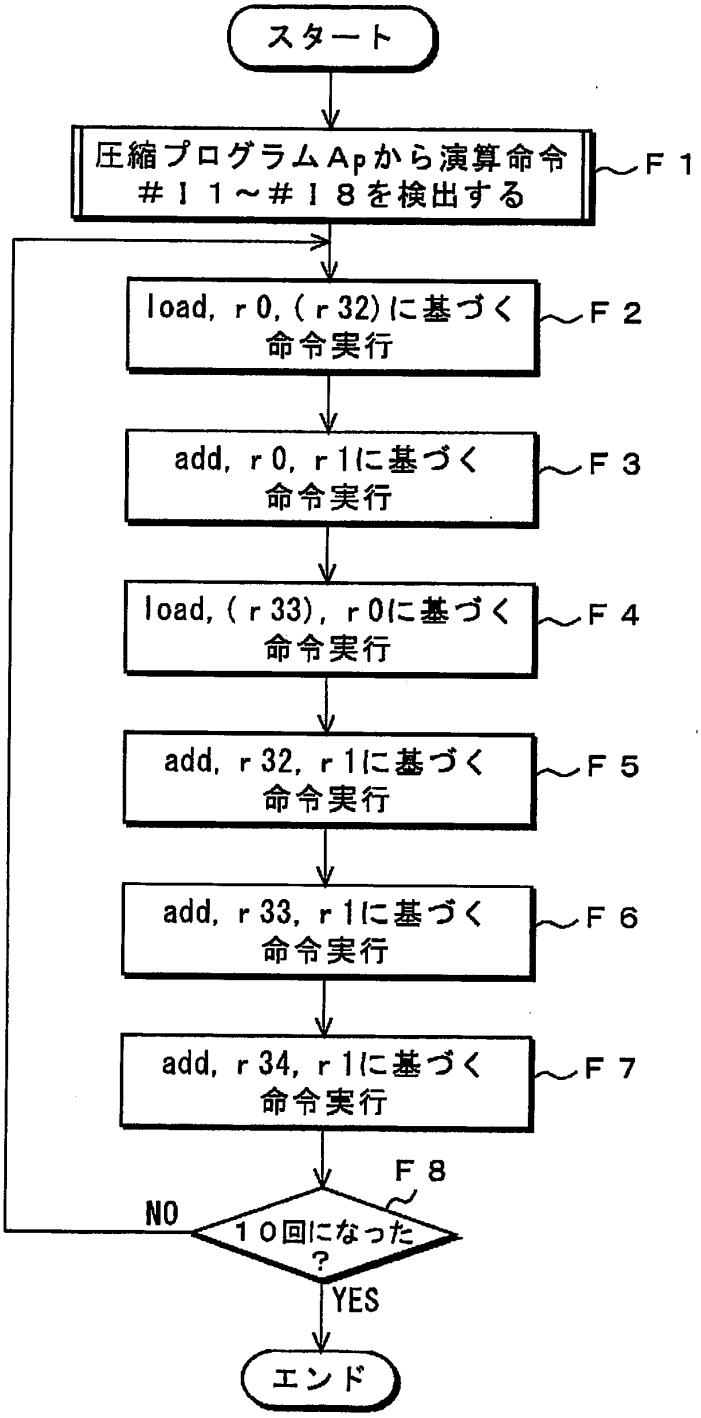
【図１２】

外部メモリ２におけるデータ格納例



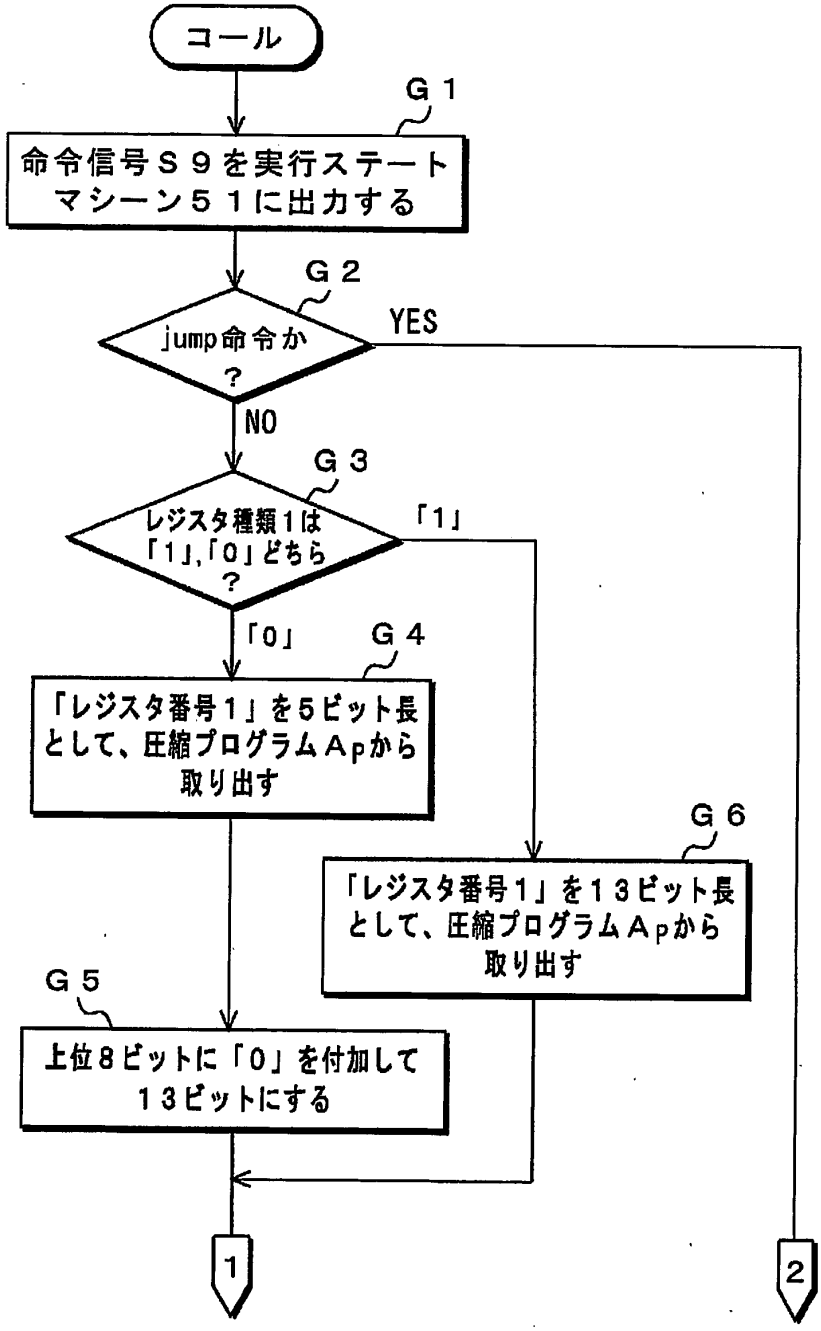
【図１３】

マイクロプロセッサ１０１における動作例



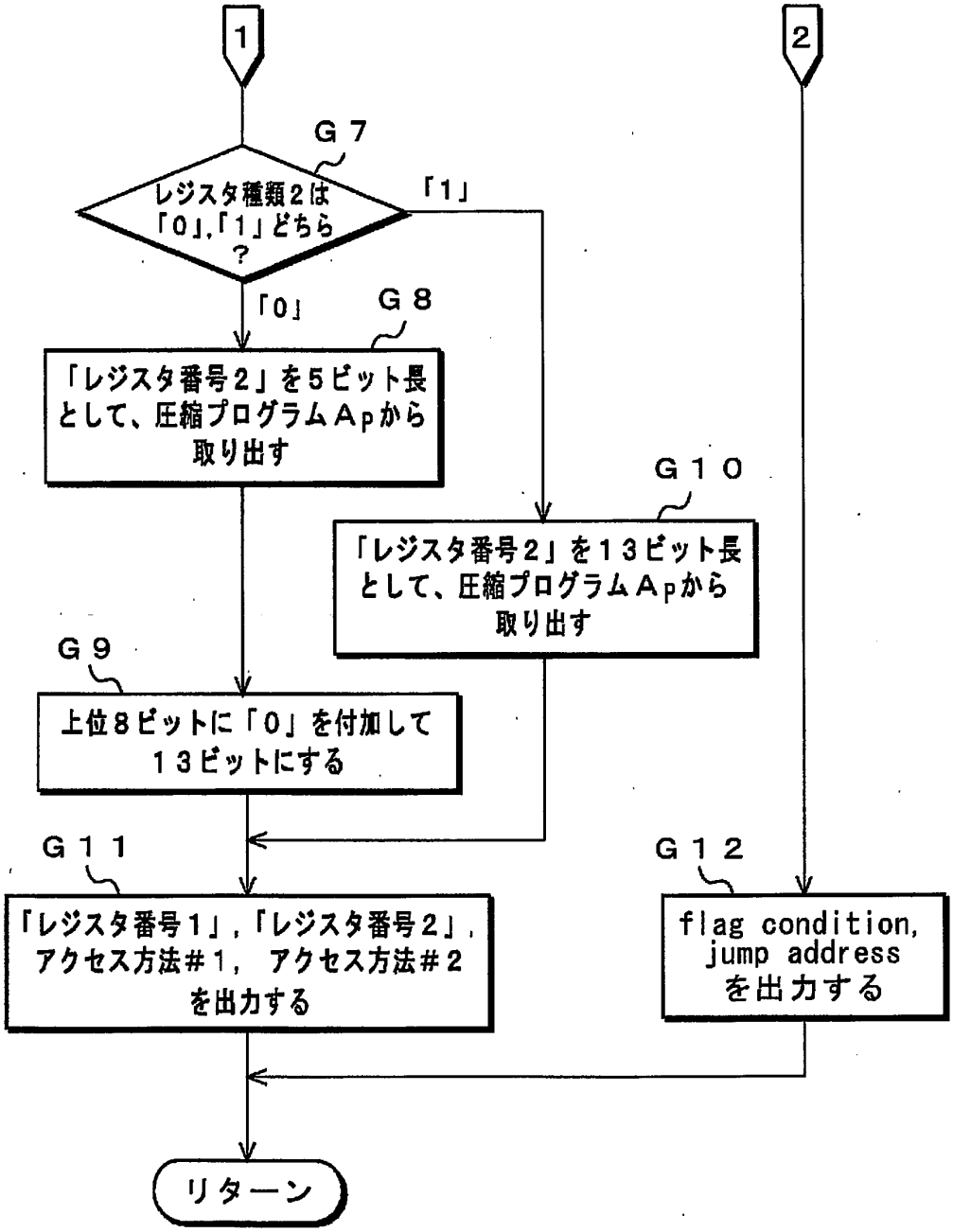
【図１４】

命令ビット復元デコーダ１３における処理例（その１）



【図１５】

命令ビット復元デコーダ１３における処理例（その２）



【書類名】　　要約書

【要約】

【課題】　レジスタの使用頻度に応じて命令の長さを可変できるようにすると共に、頻繁にアクセスするレジスタには短い長さの命令をセットできるようにし、プログラムデータを格納するＲＯＭ等のメモリ容量を低減できるようにする。

【解決手段】　レジスタｒｉを使用する頻度に基づいて当該レジスタｒｉを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムＡＰを作成するプログラム作成装置２００と、このプログラム作成装置２００で作成された圧縮プログラムＡＰを取得してレジスタ種類を解読し、レジスタ種類に基づいて当該レジスタｒｉを指定する命令ビット数を復元し、所定の命令長の命令構造に復元したプログラムに基づいて複数のレジスタｒｉを指定して任意の演算を実行するデータ処理装置１００とを備えるものである。

【選択図】　　図１

出 願 人 履 歴 情 報

識別番号　〔０００００２１８５〕

１．変更年月日　１９９０年　８月３０日
［変更理由］　新規登録
住　所　東京都品川区北品川６丁目７番３５号
氏　名　ソニー株式会社